

MSX EXTRA
LA PRIMERA REVISTA DE MSX DE ESPAÑA
N.° 36 Octubre 1987 - PVP 275 ptas. (incluido IVA)
PVP 275 ptas. Incluido IVA y sobretasa aérea Canarias
ESPECIAL SONIMAG 87
Todas las novedades del Salón
CALL XV
Las interioridades del DOS
SUPER RAMBO ESPECIAL
El último MEGA-ROM para MSX-2
MSX-2
Desvelamos los secretos de los slots.
BIT-BIT
Howard the Duck, Fernando Martín Basket, Spirits, Glass, Zanac, Uchi-Mata, Speed King, Splash, Winter Games.
TRATAMIENTO DE FICHEROS
Quicksort, la más rápida.
¡¡SENSACIONAL!!
ESPECIAL POKES DE VIDAS INFINITAS
Trucos y programas para conseguir los POKES de tus juegos preferidos.
MSX
JORGE

Diseñada para servir

al usuario de un PC. Stop

EL COMO Y EL POR QUE DE

UN STANDAR COMUN. Stop



OTRO PRODUCTO MANHATTAN TRANSFER; S.A. Stop

A la vanguardia de la prensa útil. Stop

SUSCRIBASE HOY MISMO A...

PCOMPATIBLE



UNA REVISTA FUNDAMENTALMENTE UTIL

POR EL PRECIO DE *DIEZ NUMEROS*

RECIBIRA DOCE EN SU DOMICILIO

ENVIE HOY MISMO ESTE BOLETIN DE SUSCRIPCION

Sí, deseo suscribirme a PCompatible a partir del número ..... para cuyo pago adjunto talón bancario a la orden de Manhattan Transfer, S.A., c/. Roca i Batlle 10-12, 08023. Barcelona.

Nombre y apellidos ....
Calle .... N.º ....
Ciudad .... Provincia ....
Código Postal .... Teléfono ....

Tarifas: España por correo normal Ptas. 2.750.– Europa por correo aéreo Ptas. 3.000.–
Europa por correo normal Ptas. 3.000.– América por correo aéreo USA. $ 35.–

Importante: Colocar en el sobre: Departamento Suscripciones PCompatible. NO SE ADMITE CONTRAREEMBOLSO.

Editorial

¿MSX-3?

En un número reciente de una conocida revista del sector aparecía una interesante reseña sobre la nueva generación de ordenadores MSX, en la que se llegaba a afirmar, entre otras cosas, que los nuevos MSX dejarían de llamarse así, y que estarían basados en el nuevo chip de Intel, el I-80386, abandonando de este modo el popular Z-80 con que cuentan los MSX de primera y segunda generación.
Esta noticia ha causado una verdadera conmoción entre los usuarios de MSX, ya que el cambio de chip supondría un grave problema de compatibilidad. La publicación a la que nos referíamos, soslayaba este problema con la posible inclusión de un emulador Z-80 en los nuevos aparatos.
Pero no nos molesta que se den este tipo de noticias de una forma poco seria y profesional. Lo que nos molesta es que se trate de puras fantasías, fruto de un afán de espectacularidad, de llenar una revista tan sólo con noticias espectaculares a todo color.
Aprovechando la visita realizada por Mr. Franch Berberich, director del departamento internacional de ASCII Corporation (fundadores de la norma MSX) a nuestro país le consultamos sobre este tema, desmintiéndonos tajantemente todas estas noticias.
Cierto es que desde hace unos seis meses los fabricantes del estándar se plantean el lanzamiento de nuevos ordenadores; pero lo hacen con una visión de completa continuidad: no se abandonará el Z-80, sino que se mejorarán y acelerarán los diferentes coprocesadores (VDP, PSG, etc). De este modo se consiguen mejores prestaciones sin perder un ápice de compatibilidad.
Por último decir que en algo sí acertaron (pura suerte quizás), ya que probablemente los nuevos ordenadores no se llamen MSX-3. Las razones son de lo más inocente. Las letras MSX (provenientes en principio de MicroSoft eXtended) son muy difíciles de pronunciar en japonés, cosa que, comercialmente es un handicap que debe solucionarse en los nuevos aparatos.
Prometemos ampliar esta información en siguientes números de nuestra revista, así como las esperanzadoras noticias que Mr. Berberich nos ofreció acerca del futuro del estándar.



SUMARIO

AÑO III N.º 36 OCTUBRE 1987
P.V.P. 275 ptas. (Incluido IVA y sobretasa aérea Canarias)
Aparece los días 15 de cada mes.

MSX EXTRA ES EDITADA POR MANHATTAN TRANSFER, S.A.
Director Ejecutivo: Birgitta Sandberg.
Redactor Jefe: Javier Guerrero.
Redactores: Willy Miragall, Silvestre Fernández, Rubén Jiménez y Carles P. Illa
Colaboradores: Angel Toribio, Fco. Jesús Viceyra, Joaquín López.
Departamento de Programación: Juan C. González. **Diseño:** Félix Llanos.
Grafismo: Juan Núñez, Jordi Jaumandreu, Carles Rubio. **Suscripciones:** Silvia Soler. **Redacción, Administración y Publicidad:** Roca i Batlle, 10-12. 08023 Barcelona. Tel.: (93) 211 22 56. Télex: 93377 TXSE E.
Depósito legal: M-7389-1987.
Fotomecánica y Fotocomposición: Ungraf, S.A. Pujadas, 77-79. 08005 Barcelona.
Imprime: Grefol, Polig. II Lafuensanta Parc. 1 Móstoles (Madrid)
Distribuye: GME, S.A. Plaza de Castilla 3, 15.º E. 2. 28046 Madrid

¿DIBUJAR CON DATA?

Soy un novato en esto y lo que más me gusta hacer con mi MSX es hacer gráficos. También me gustaría saber dibujar con el mandato DATA pero por muchas pruebas que haga no sale nada en la pantalla.

Jordi Montañés
BARCELONA

Lamentamos tener que comentarte que la sentencia DATA no tiene nada que ver con los gráficos. Es cierto que muchos programas gráficos utilizan esta sentencia; pero también se utilizan GOTO, IF...THEN... y FOR...NEXT sin que nadie las asocie con los gráficos.

Tendrás que contentarte con los sistemas habituales de realizar gráficos en los MSX.

COMPATIBILIDAD MSX-2

En la sección Input/Output del número 31 de MSX-Extra apareció una carta en la que Sergio Mojón Suárez decía tener problemas de carga con su ordenador Sony HB-F9S. A mí me sucedió lo mismo al comprar dicho aparato; pero tras ir al concesionario SONY me confirmaron que sólo se necesita introducir un POKE antes de cargar el juego de la cinta. Este POKE es:

POKE 65536,170

Mónica Ortega Fernández
MALAGA

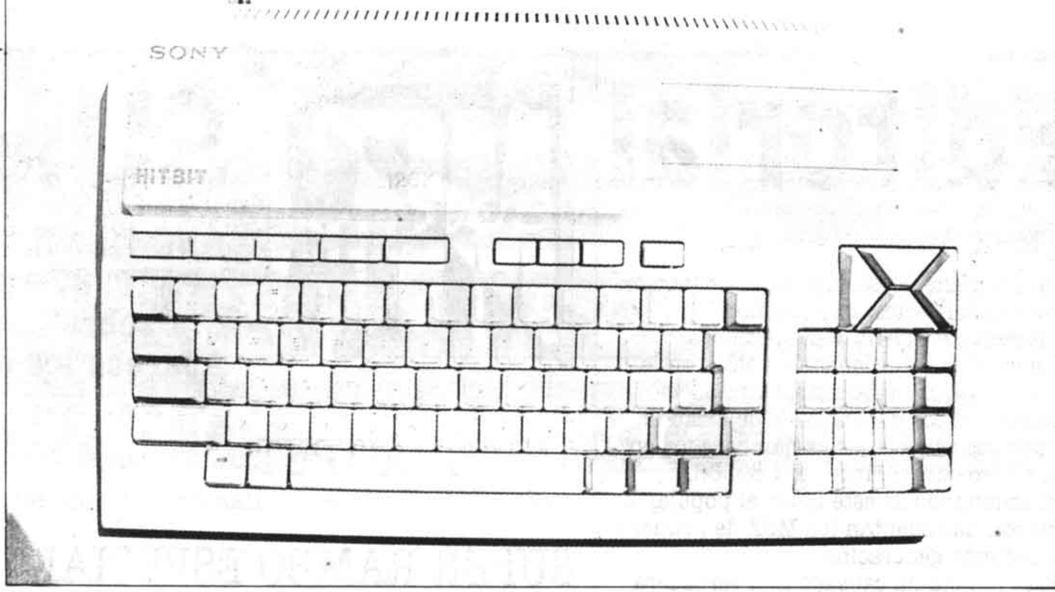
Te agradecemos que nos hayas hecho llegar tu carta; pero creemos que los problemas de carga que surgieron a nuestro comunicante eran de otra índole por las explicaciones que nos daba en su carta.

De todas formas hemos creído interesante publicar tu carta, ya que muchos otros usuarios tienen tu mismo problema.

Lamentablemente el POKE que nos incluyes funciona sólo en ciertas ocasiones. Ya hemos hablado de estos POKES indicando que no son fiables al 100 %. Añadimos otro POKE para aquellos usuarios de MSX-2 que no consigan cargar sus problemas.

POKE 65535,255

Estos POKES funcionarán o no dependiendo del programa a cargar, así como del aparato en que estemos trabajando.



Sony HB-F9S

PROGRAMACION DE JUEGOS

Tengo entendido que en los juegos se utiliza mucho el lenguaje máquina para darles más rapidez. ¿Para qué más?

Quisiera especializarme en programas de juegos. ¿Dónde podría (si es que existe algún lugar) estudiar aquí en Barcelona o alrededores?

¿En qué consiste vuestro test de listados?

Joaquín Márquez Carrizo
Barberá del Vallés (BARCELONA)

El lenguaje ensamblador (C.M.) se utiliza en los juegos debido a sus dos propiedades fundamentales: rapidez y economía de memoria. Estas dos razones hacen que éste sea el lenguaje preferido por los programadores de juegos, así como por aquellos que diseñan y trabajan con periféricos especializados (MIDI, Robótica, etc).

Lamentablemente no conocemos de ningún centro especializado en informática que tenga cursos de programación de videojuegos. Existía uno en Barcelona (especializado en MSX) que incluía en sus cursos ensamblador, juegos, y otros interesantes aspectos pero ha cerrado sus puertas a la enseñanza desde principios de este verano.

Respecto al test de listados, se trata de una rutina que emite un listado de números de control. Una vez hemos copiado un programa de alguna de nuestras revistas hemos de aplicarle el programa Test de Listados. Este programa nos entregará una cifra de control. Si esta cifra coincide con el TOTAL publicado en la revista significa que el programa ha sido correctamente transcrito. En caso contrario cada línea del programa tiene su propia cifra de control, con lo que podemos averiguar si está bien transcrita desde la revista.

TECLAS LOCAS

¿Qué ocurre cuando pulso al mismo tiempo las teclas CODE+GRAPH+SHIFT y posteriormente CTRL? Cuando lo hago en el menú del principio se me pone la pantalla de un solo color y si hago SCREEN 0 y doy LIST me aparece un listado extraño y me da RESET. Si hago esto en BASIC me salen Ok hasta que deje de pulsar CTRL.

¿Debo preocuparme? ¿Por qué sucede esto?

Santiago Martínez
MADRID

Pese a que los MSX son aparatos de muy alta calidad no son perfectos. Debido a la peculiar estructura del chip que controla el teclado, se producen ciertos «errores» al pulsar numerosas teclas simultáneamente. Dado que esto no suele ocurrir con demasiada frecuencia se pasó por alto en los controles de calidad y ahora nosotros, sufridos usuarios del estándar nos encontramos con que nuestros ordenadores fallan ante esta eventualidad.

Hemos probado la combinación de teclas que nos comentas en varios ordenadores (todos ellos SONY) y no hemos detectado que ocurra nada especial. Sin embargo no te preocupes. Por ejemplo si pulsamos al mismo tiempo las teclas SHIFT+comillas+"A" aparecerá ante nosotros el mensaje CLOAD".

Es algo que difícilmente podemos ayudarte a solucionar. Debes aprender a convivir con este tipo de errores sin preocuparte. Simplemente evita el pulsar más de 3 teclas simultáneamente en tu ordenador. ¡Y no te preocupes en absoluto! Por muy raras que sean las cosas que haga tu ordenador en estas condiciones, son totalmente inócuas para su circuitería. Como máximo puedes perder los datos contenidos en la memoria en ese momento.

PRINT USING

Estoy haciendo un programa de gestión para llevar la contabilidad de un pequeño negocio y el problema

me sale al imprimir las cifras, ya que las pone de izquierda a derecha y no de derecha a izquierda como yo quisiera para que cuadren bien los números. Tengo entendido que esto se soluciona con «PRINT USING» pero no sé cómo utilizarlo.

Manuel Lara Pérez
MALAGA

Desgraciadamente una instrucción tan importante como PRINT USING se encuentra casi olvidada en los manuales de BASIC.

Esta instrucción se utiliza como un PRINT normal; pero los datos que queramos imprimir pueden formatearse a nuestro gusto. Por ejemplo, si queremos imprimir números enteros en un espacio de 8 caracteres y que queden alineados a la derecha deberemos hacer:

PRINT USING "######## ";Num

Si queremos que el número aparezca con dos decimales haríamos:

PRINT USING "#####.##";Num

Existen muchas más posibilidades para esta potente instrucción; pero comentarlas todas sería demasiado extenso. Esperamos que estos dos ejemplos basten para tus propósitos.

GRABACION DE PANTALLAS

En el programa de diseño gráfico "DRAW AND PAINT" al hacer el dibujo y luego al grabarlo funciona bien; pero al reproducirlo sólo se puede ver cuándo el diseñador está en memoria.

La pantalla de presentación se carga en las direcciones:

INICIO: &H0000
FIN: &H37FF
EJEC: &H0000

Una vez cargado, en vez de ejecutarse se hace un RESET. Me han dicho que puedo cambiar las posiciones de memoria. ¿Cómo puedo hacerlo?

Rolo Padonani
GRANADA

Ningún programa ni gráfico puede cargar en las direcciones de memoria que nos comentas de la RAM principal, ya que esta memoria está ocupada por la ROM BIOS (si se utiliza el BASIC) o por el sistema de interrupciones no enmascarables (en cualquier otro caso).

Suponemos por tanto, que tales direcciones corresponden a la VRAM, y que el dibujo es cargado por el programa directamente en la VRAM.

Si utilizas unidad de disco es muy fácil conseguir esta carga desde el BASIC, con la instrucción BLOAD "nom",S.

Pero si trabajas desde el cassette deberás cargar la pantalla en memoria principal y de ahí, con una pequeña rutina en ensamblador, pasar la RAM a VRAM.

Para cargar la pantalla en memoria principal (por ejemplo en la dirección &H9000) deberás hacer:

BLOAD "nom",&H9000

sólo queda pasar la RAM a VRAM con una rutina en C.M. o bien con un bucle en BASIC.

SACAR JUGO A LOS PROGRAMAS

¿Para qué sirve la BASE(10)? ¿Qué se almacena ahí y cómo? ¿Para qué sirve si ya tenemos las bases 11 y 12?

Tengo un par de programas con rutina inteligente hechos por mí y quisiera saber cómo sacarles jugo para poder comprarme una unidad de discos.

Enric Carrera i Gallart
LLEIDA

La variable del sistema BASE(10) almacena la TNP (tabla de números de los patrones) en SCREEN 2. Aunque al usuario final no se lo parezca, el modo SCREEN 2 funciona de forma similar a como lo hacen los modos de texto, ya que la pantalla está formada por caracteres que se van redefiniendo a la hora de realizar los dibujos.

La tabla cuya dirección se indica en BASE(10) contiene los caracteres que conforman la pantalla gráfica, y gracias a ella se pueden conseguir efectos muy espectaculares, scrollings parciales de la pantalla, etc.

Te recomendamos, si quieres saber algo más sobre el tema, que leas el artículo aparecido en el número 32-33 de nuestra revista: «Reprogramación de caracteres en SCREEN 2».

Respecto a «sacarle jugo» a tus programas; existen muchas formas de hacerlo. La más lógica es la de vender dichos programas. Si los programas estuvieran hechos en C.M. y tuvieran una buena calidad podrías intentar editarlos en cinta por medio de alguna empresa de Soft nacional, como DINAMIC o Manhattan Transfer (a través del MSX-Club de cassettes). Si los programas están hechos en BASIC es difícil que se editen en cinta ya que el mercado está saturado de este tipo de programas.

Puedes, no obstante, enviarnos tus programas, que si tienen una buena calidad podrán ser publicados en nuestras revistas, con la consiguiente retribución económica. ¡Es sólo una idea...!

FALLOS EN DISCO

Tengo una unidad de discos de simple cara SONY y un ordenador TOSHIBA HX-10 y quisiera que me resolvieran unos problemas que me han surgido.

¿Puedo utilizar diskettes vírgenes 2DD sin problemas?

A veces me aparece DISK FULL aunque quede mucho espacio en el disco, y los programas que aparecen en el directorio no se ejecutan correctamente. ¿A qué puede ser debido?

Bautista Enrique
Burriana (CASTELLON)

Pese a que es un derroche innecesario no existe ninguna contraindicación por utilizar discos 2DD en tu unidad. Te recomendamos sin embargo, por razón de economía, que utilices discos de una sola cara en tu unidad de discos.

Respecto al error que nos comentas (ocurrido repetidamente en diferentes discos) es debido a la sobreescritura de datos en el directorio del disco. Esto puede tener su origen en dos puntos diferentes.

1.— Estás grabando en tus diskettes programas comerciales no preparados para ser almacenados en disco. Estos programas utilizan las mismas zonas de memoria que el interfaz de la unidad de disco, produciéndose de este modo un conflicto que puede llegar a provocar la sobreescritura del directorio.

2.— Si no estás incluyendo en tu disco este tipo de programas, sino programas en BASIC, o programas preparados específicamente para el disco, se debe tratar, sin duda, de un error en la unidad de disco que deberá ser solucionado por el servicio técnico correspondiente.

VOCES HUMANAS EN MSX

Quisiera rectificar lo que habéis dicho sobre la obtención de voces humanas en el número 11 de vuestra revista, ya que hay un juego (Valkyr) para MSX en que se emiten voces humanas, lo cual quiere decir que el chip de sonido del MSX puede emitir voces humanas.

José Antonio Fernández Rodríguez
LA CORUÑA

Agradecemos tu rectificación y reconocemos nuestro craso error. Los MSX pueden perfectamente emitir voces humanas, y así también cualquier otro sonido e incluso funcionar como sintetizador monofónico.

A ti tenemos que corregirte y hacerte saber que no es el chip de sonido el encargado de estos efectos, sino un bit del PPI (Port Programable de Interfaces) que controla directamente el altavoz. El chip de sonido, por su especial forma de programación es incapaz de realizar estos efectos.

Sin embargo, y gracias a este bit del PPI se puede enviar directamente al altavoz cualquier sonido previamente digitalizado: una voz, un grito, o cualquier otro

efecto que se desee. Esperamos tratar pronto este tema e incluir una rutina que os permita utilizar tales efectos en vuestros propios programas.

ADAPTADOR CCG-MSX

Quisiera que me informaráis si el adaptador de MSX para SVI-328/318 se comercializa en Madrid y en el caso de que así sea, dónde puedo conseguirlo y cuál es su precio.

También querría saber cuáles son las líneas que faltan al programa Talismán del número de Abril.

Juan Enrique Pérez Muñoz
MADRID

Según las noticias de que disponemos en este momento hemos de decirte que el adaptador CCG para convertir los SVI-318 y SVI-328 en MSX sólo se comercializa desde Baracaldo. Sin embargo puedes dirigirte a CCG por correo, ya que es éste el medio por el cual comercializan su adaptador. El precio del mismo el 1 de Enero de este año era de 7.840 ptas (IVA incluido) y el del subsistema de disco opcional de 1.200 ptas.

La dirección de CCG S&H es la siguiente.

C.C.G. S&H
Larrasolo, 13-4B
48902-BARACALDO (VIZCAYA)
Tno: (94) 462 46 52
(94) 440 29 99

Respecto al programa TALISMAN, desgraciadamente hubo un error de fotocomposición, y fueron equivocadas dos páginas del programa. Dichas páginas se incluyeron en el siguiente número de MSX-Extra (número 31 - Mayo).

FE DE ERRATAS

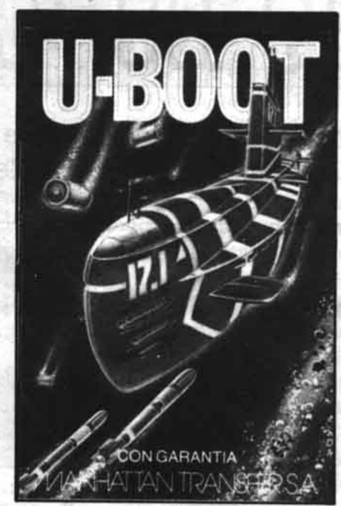
Andoni Rego Etxebarria, creador del programa VERBOS para MSX-2, que apareció publicado en nuestro número 34 nos avisa de que en el listado publicado existe un pequeño error. La línea 1280 de este programa debe quedar así:

```
128Ø IF G$="12:ØØ:ØØ"
     THEN PLAY
"v1ØL64ade"
```

IDOS A msxclub de CASSETTES



KRYPTON. La batalla más audaz de las galaxias en cuatro pantallas y cuatro niveles de dificultad. Un juego cuya popularidad es cada vez más grande entre los usuarios del MSX. PVP. 500 Ptas.



U-BOOT. Sensacional juego de simulación submarina en la que tienes que demostrar tu pericia como capitán de un poderoso submarino de guerra. Panel de mandos, sonar, torpedos, etc. PVP. 700 Ptas.



QUINIELAS. El más completo programa de quinielas con estadística de la liga, de los aciertos, etc. e impresión de boletos. Acertar no siempre es cuestión de suerte. PVP. 700 Ptas.



SNAKE. Entretenido y muy divertido juego en el que Snake procura comer unos números que la engordan. Tanto las murallas que la rodean como su larga cola pueden ser mortales para ella. PVP. 600 Ptas.



EL SECRETO DE LA PIRAMIDE. Atrevido juego de aventuras a través de los misterios y peligros que encierran los laberínticos pasillos de una pirámide egipcia. ¡Atrévete si puedes! PVP. 700 Ptas.



STAR RUNNER. Conviértete en el audaz piloto interestelar y lucha a muerte, a través del hiperespacio, contra las defensas del tirano Daurus. Dos pantallas y cinco niveles de dificultad. PVP. 1.000 pts.



FLOPPY, El Preguntón. Un verdadero desafío a tus conocimientos de Geografía e Historia española. Floppy no perdona y te costará mucho superarlo. PVP. 1.000 Ptas.



MAD FOX. Un héroe solitario es lanzado a una carrera a vida o muerte por un desierto plagado de peligros. Conseguir el combustible para sobrevivir es su misión. Diez niveles de dificultad. PVP 1.000 pts.

Si quieres recibir por correo certificado estas cassettes garantizadas recorta o copia este boletín y envíalo hoy mismo:

Nombre y apellidos: ....

Dirección: ....

Población: .... **CP** .... **Prov.** .... **Tel.:** ....

☐ KRYPTON .... Ptas. 500,–	☐ EL SECRETO DE LA PIRAMIDE .... Ptas. 700,–	☐ MAD FOX .... Ptas. 1.000,–
☐ U BOOT .... Ptas. 700,–	☐ STAR RUNNER .... Ptas. 1.000,–	☐ VAMPIRO .... Ptas. 800,–
☐ HARD COPY .... Ptas. 2.500,–	☐ TEST DE LISTADOS .... Ptas. 500,–	☐ SKY HAWK .... Ptas. 1.000,–
☐ LORD WATSON .... Ptas. 1.000,–	☐ MATA MARCIANOS .... Ptas. 900,–	☐ TNT .... Ptas. 1.000,–
☐ LOTO .... Ptas. 900,–	☐ DEVIL'S CASTLE .... Ptas. 900,–	
☐ SNAKE .... Ptas. 600,–	☐ FLOPPY .... Ptas. 1.000,–	

Gastos de envio certificado por cada cassette .... Ptas. 70,– Remito talón bancario de Ptas. .... a la orden de Manhattan Transfer, S.A.

ATENCION: Los suscriptores tienen un descuento del 10% sobre el precio de cada cassette.

IMPORTANTE: Indicar en el sobre MSX CLUB DE CASSETTES. ROCA I BATLLE, 10-12 BAJOS. 08023 BARCELONA

Para evitar demoras en la entrega es imprescindible indicar nuestro nuevo código postal.

NUESTRAS CASSETTES NO SE VENDEN EN QUIOSCOS. LA UNICA FORMA DE ADQUIRIRLAS ES SOLICITANDOLAS A NUESTRA REDACCION. ¡NO SE ADMITE CONTRA REEMBOLSO!

SUPER RAMBO ESPECIAL

Los MEGAROM comienzan a desembarcar en nuestro mercado. En este caso, este sensacional juego se basa en las aventuras de RAMBO en su camino por liberar a un prisionero de los vietnamitas.

¡Todo en orden! Sobrevolamos la zona de lanzamiento... Puede lanzarse cuando lo desee... Así empieza este espectacular juego. RAMBO sobrevuela la zona enemiga y se lanza en paracaídas para rescatar al único superviviente de los campos de prisioneros vietnamitas. En su camino sólo encontrará ingentes cantidades de enemigos que, sin embargo, son su único método para conseguir el armamento necesario para llevar a buen término su difícil misión.

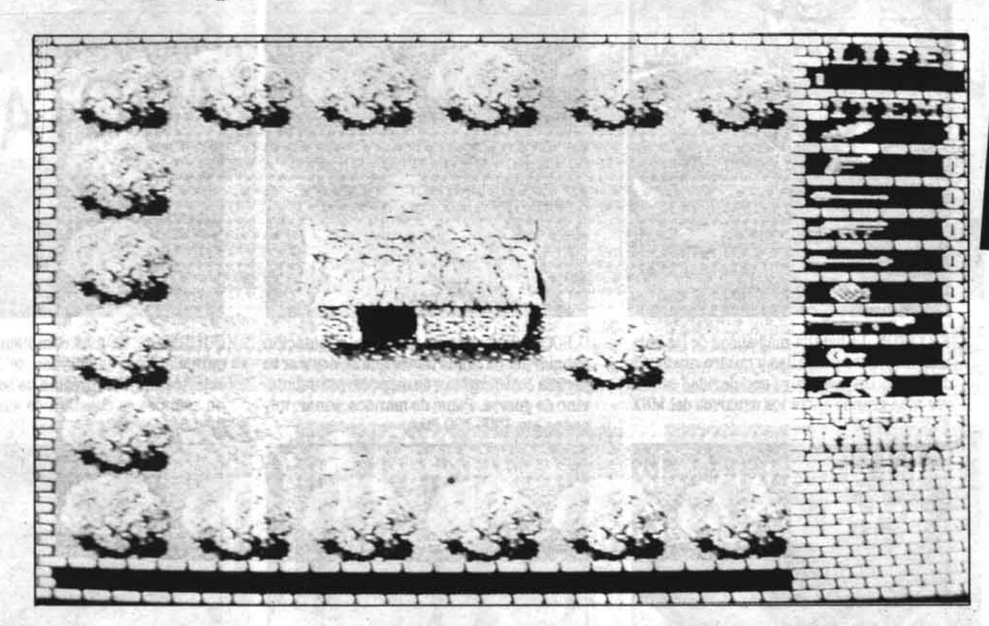
Pero antes de entrar de lleno en el juego comentemos algo sobre el soporte físico del mismo.

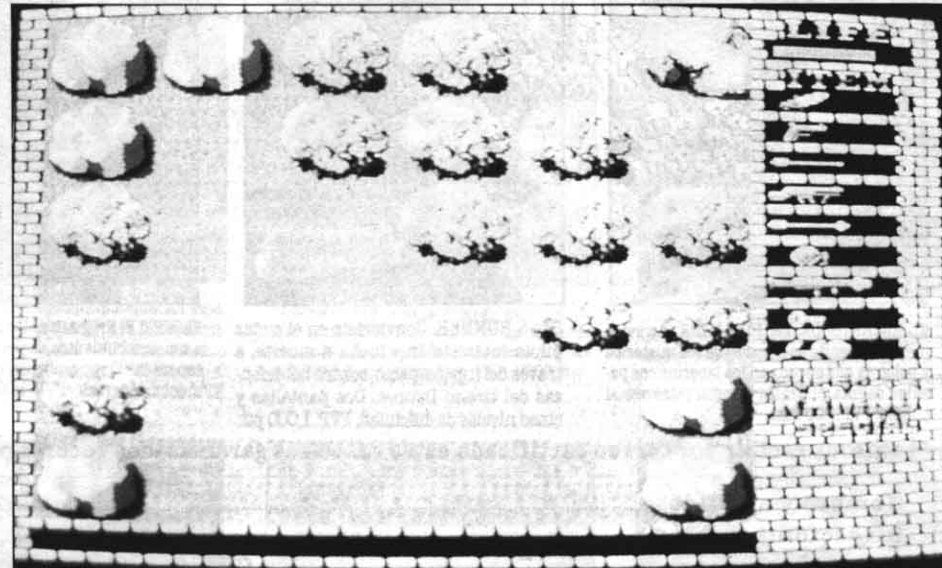
SUPER RAMBO

SUPER RAMBO son en realidad dos juegos, dos cartuchos de ROM, para ser exactos. El primero es una versión para los MSX de primera generación, mientras que el segundo, SUPER RAMBO ESPECIAL, es un MEGAROM que sólo funcionará en los MSX de segunda generación.

Existen varias diferencias entre ellos. La primera y más palpable es la enorme diferencia existente en la calidad de los gráficos empleados en ambas versiones. Los gráficos empleados en la versión ESPECIAL (sólo MSX-2) tienen un excelente grado de realismo. En cambio, la versión MSX-1, para subsanar su menor calidad gráfica, ha incluido más peligros en el juego (arañas y serpientes venenosas entre otros), con lo que el juego se hace algo más interesante.

Para centrarnos adecuadamente hablaremos, a partir de ahora, de sólo una de las versiones, SUPER RAMBO ESPECIAL, si bien muchas de los puntos comentados servirán también para SUPER RAMBO (MSX-1).





Estas dos pantallas situaciones típicas en el juego: La cabaña de prisioneros y el bosque.

SUPER RAMBO ESPECIAL

Antes de comenzar el juego queremos recomendaros la utilización del mapa de este juego, publicado en el número de Septiembre de nuestra revista hermana MSX-CLUB en forma de póster en color. Gracias a este mapa podréis localizar rápidamente los pasajes secretos que os conducen por entre los setos a diferentes depósitos de armas, conocer el lugar en que se encuentra el arma que esteis buscando, etc.

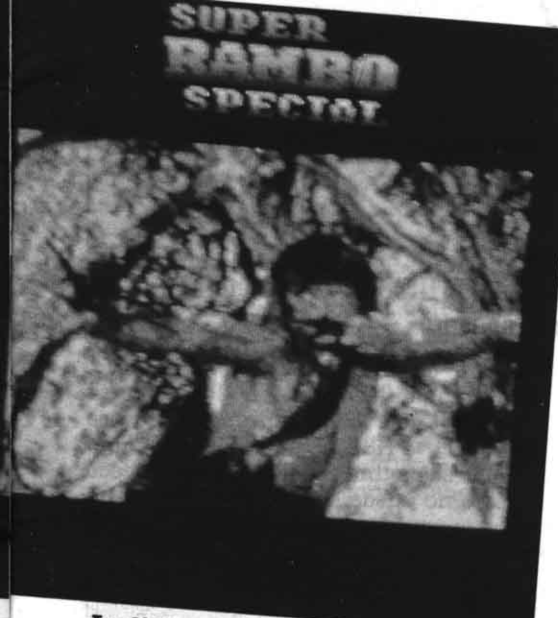
Nada más conectar el cartucho a vuestro MSX-2 aparece en la pantalla una excelente digitalización de RAMBO, que sirve de presentación al juego.

El objetivo de la misión es recorrer más de 250 pantallas hasta localizar al prisionero y luego huir con él hasta el helicóptero que se encuentra en un claro de la selva preparado para la huida.

Pero este camino no es nada fácil. Existe un gran número de soldados enemigos acuartelados en esta zona, defendiendo con uñas y dientes unas pequeñas edificaciones de caña de bambú (realizadas con unos gráficos soberbios). Dentro de estas edificaciones se encuentran (además de cinco o seis nuevos enemigos) las armas que RAMBO precisa para terminar la misión.

Para complicar más aún ésta, muchas de las cabañas de caña están cerradas con gruesos candados y para poder entrar se necesitarán de unas llaves que, cómo no, se encuentran esparcidas a todo lo largo y ancho de la selva que compone el escenario de este juego.

Y aún hay más. A lo largo de toda la



La digitalización de Rambo que sirve de presentación al juego para MSX2

aventura encontraréis varios parterres sembrados con «lindas florecitas». Existen dos tipos de flores: las flores azules, medicinales, recompensarán a RAMBO con una cantidad extra de vida, mientras que las rojas y blancas, venenosas, reducirán considerablemente ese nivel de energía.

A los enemigos los distinguiréis fácilmente por el color de sus uniformes, y según sea éste, dispondrán de un arma determinado: revólver, metralleta, o granadas y cohetes (son los más peligrosos). En cuanto a las armas de RAMBO, dispone de las siguientes: cuchillo, pistola, metralleta, flecha, granada, cohete y bazooka.

Pistolas, metralletas, flechas son armas muy aparentes; pero con poca utilidad. Para poder utilizarlas hemos de estar en línea de tiro con el enemigo, que por esa misma razón no duda en dispararnos a nosotros.

Sin duda, el arma más eficaz es el cuchillo. Sólo hay que acercarse sigilosamente al enemigo y atacarle cuando esté desprevenido.

Por último granadas, cohetes y bazookas son fundamentales para el éxito de la misión, ya que nos permitirán arrasar cabañas que nos cierran el paso, o destruir rocas molestas. Hay que decir que, personalmente, preferimos el bazooka, ya que es el arma con mayor poder destructivo.

Sólo una cosa más respecto a las armas. Estas se acumulan mientras no las utilicemos, por lo que RAMBO puede acabar portando un envidiable arsenal. De igual forma se acumulan las flores azules, que podremos utilizar cuando nos sean necesarias, y las llaves que nos permitirán abrir ciertas cabañas.

EMPECEMOS EL JUEGO

Primera fase: Al iniciar el juego nos encontramos en medio de la selva, contando con un cuchillo como única arma. Lo primero que debemos hacer es aprovisionarnos del arsenal adecuado. Para ello lo más interesante es bajar a la pantalla inferior y de allí hacia la izquierda. Recorriendo unas cuantas pantallas arriba y abajo llegaremos a una cabaña que contiene el bazooka. Este arma es totalmente necesaria, ya que sin ella no podremos llegar hasta nuestro primer campamento.

Antes de llegar a la cabaña del bazooka notaremos que un extraño aliado se ha unido a nosotros, disparando a todos nuestros enemigos. ¡Intentad conservar su ayuda todo el tiempo posible, haciendo que se aparte del fuego enemigo!

Sólo al final de esta primera etapa encontraremos nuestro camino cerrado por unas rocas. Es el momento de utilizar el bazooka. Tras destruir esas rocas encontraremos una cabaña. Al entrar en ella observaremos un código en la parte inferior de la pantalla. A partir de ahora podemos empezar nuestras siguientes partidas en este punto con sólo introducir el código (además empezamos con un interesante arsenal).

CLAVE 1: 10A09A38

Segunda fase: La segunda fase es bastante similar a la primera, con una salvedad. Encontraremos dos puntos en los que las flores venenosas nos cierran el paso. En el primero podemos abrirnos paso destruyendo con el bazooka una cabaña de bambú. Para pasar al segundo plantel de flores lo mejor es utilizar una flor azul. Esta flor nos dará la suficiente energía como para atravesar el plantel y llegar a nuestro segundo campamento, donde se nos dará una nueva clave.

CLAVE 2: 04K14042

Tercera fase: La tercera fase es bastante corta y sencilla; pero necesitamos una buena provisión de flores azules. Es además muy importante encontrar el pasaje secreto (entre setos que no son tales), ya que nos evitará gastar una flor azul.

En la última parte de esta fase encontramos una enorme extensión de plantas blancas y azules; pero como están situadas de forma alterna, podemos traspasarlas sin ningún problema, ya que la energía que nos reste una flor venenosa, nos la recuperará la flor azul.

CLAVE 3: 03M16041

Cuarta fase: El río. En este río tropical, lleno de fango y rocas, debemos luchar contra la corriente. Existe, sin embargo, un método muy sencillo para atravesarlo sin problemas. Basta con dirigirse a la derecha y apretar todo lo rápida que se pueda el botón de disparo. Podéis también apoyaros en las rocas que sobresalen en el curso del río.

CLAVE 4: 05U01K42

Quinta fase: Sin duda la más sencilla de toda la aventura. Basta con ir siempre hacia arriba, y localizar la cabaña en la parte izquierda de nuestro camino. La entrada se halla cerrada por unas rocas, que podemos destruir con nuestro querido bazooka, y tras ellas un pasaje secreto entre los setos nos llevará al campamento.

CLAVE 5: 03K12M41

Sexta fase: Nada más abandonar nuestro quinto campamento encontraremos un mapa (que sirve de muy poco ya que es prácticamente ininteligible). Descendiendo otro gran trecho llegamos a una zona con setos. Hay que encontrar entre ellos un pasaje secreto que nos lleva a un claro. Una vez aquí, con el bazooka, podremos abrirnos paso hasta una cabaña enemiga y, un poco más abajo, nuestro sexto campamento.

CLAVE 6: 04K29K38

Séptima fase: Superar la séptima fase es sólo cuestión de bazooka. Hay que localizar también un gran número de pasajes secretos, que nos conducirán poco a poco a la prisión. Una vez allí, y tras utilizar una llave para abrirla, podremos rescatar al prisionero.

El prisionero, una cosa raquítica que nos sigue a todos lados, es el principal objetivo de nuestra misión. No hemos de exponerlo, por ningún motivo, al fuego enemigo.

CLAVE 7: 03S21K40

Octava fase: Una fase de puro trámite, en que nos tenemos que limitar a seguir los pasillos adecuados. No encontraremos ningún obstáculo importante.

CLAVE 8: 12F04M29

Novena fase: Esta última fase es un verdadero laberinto, y se agradece más que nunca disponer del mapa publicado en nuestra revista hermana MSX-CLUB. Tras encontrar la salida a tal laberinto y destruir una cabaña que nos estorba el paso, llegamos a nuestro noveno y último campamento.

CLAVE 9: 10Y20T31

Final: Al lado de nuestro campamento encontramos un helicóptero, listo para nuestra fuga. Hemos de montar en él (con el prisionero) y huir hacia la zona inferior derecha del juego. Sobrevolaremos con el helicóptero numerosas pantallas; pero ahora ya satisfechos del trabajo realizado, y sin ningún peligro para nuestras vidas.

Tras alcanzar la pantalla inferior derecha aparece el típico mensaje THE END, en este caso acompañado por una triunfal tonada y por numerosas banderas americanas (un detalle muy acertado, dado el carácter del personaje).

Terminamos aquí el comentario de este sangriento juego, lleno de emoción y acción en sus más de 250 pantallas. Sin duda, un MEGAROM de excepción.

INTERIORIDADES DEL "DOS"

Este mes, tal y como prometimos, entraremos a fondo en las interioridades del "DOS", basándonos en la relación de funciones suministradas en el número pasado.

LA LENGUA DE SHAKESPEARE

No hace falta entender las obras de Shakespeare en su lengua natal para defenderse con los ordenadores. Sin embargo, los que creen que memorizar los vocablos correspondientes a las palabras clave del BASIC es todo lo que necesitan saber de inglés son unos optimistas. En todos los estudios técnicos superiores se incluye al inglés como asignatura obligatoria y la informática no es una excepción.

Es importante que cualquiera que piense dedicarse a la programación tenga una cierta base en esta lengua y aún más, es bueno que adquiera la costumbre de nombrar a sus procedimientos y de añadir comentarlos en inglés.

Tarde o temprano debemos presentar alguno de nuestros listados a alguien que no conoce el castellano hemos de pelearnos con algún manual en inglés.

Sin ir más lejos, corre por ahí un libro excelente sobre MSX2, escrito... ¡en japonés! Afortunadamente los autores han tenido el acierto de incluir pequeños comentarios a los listados en inglés. Muy escuetos, por cierto, pero suficientes.

Todo esto viene a cuento por el hecho de que son muchas las palabras construidas con siglas y abreviaturas que forman términos relacionados con las interioridades del DOS. La similitud de los vocablos y la cantidad de éstos hace que sea fácil perderse si no se tiene una pequeña base de inglés.

Empecemos sin más, con el aluvión de siglas.

DMA Y FCB

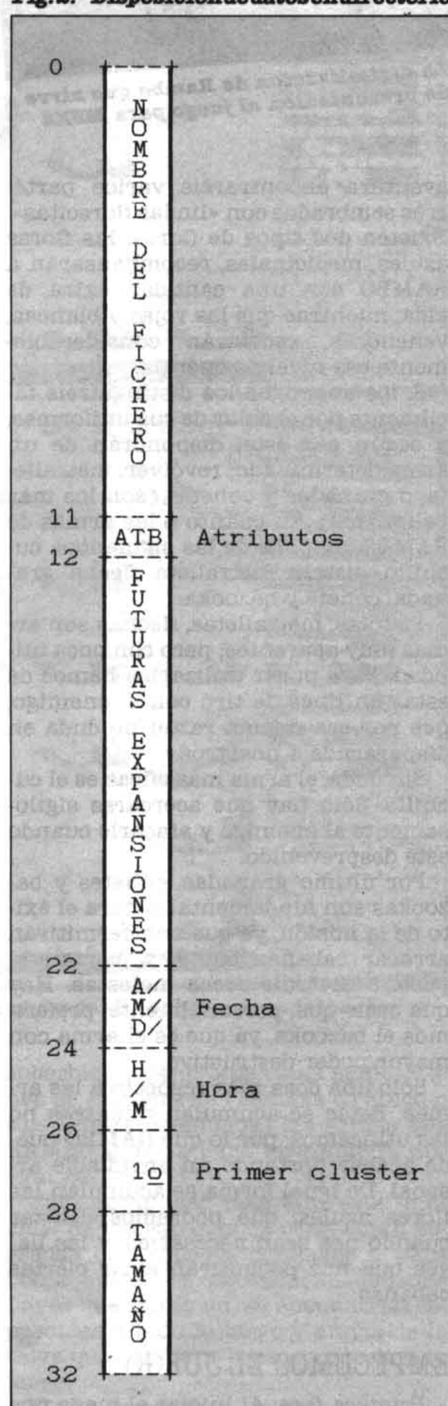
El DMA (Direct Memory Access) es la zona de comunicaciones reservada para la entrada y salida de datos hacia el disco. Con estas siglas se designa el procedimiento por el cual un dispositivo lee o escribe en la memoria sin contar con la CPU. En realidad su empleo en microordenadores es una reliquia del pasado, puesto que en la actualidad suele ser la CPU la que se encarga de la transferencia de los datos. Aún así el nombre ha perdurado.

Nosotros designaremos por DMA la posición de memoria en la que se empieza a realizar la transferencia. Es muy importante tener en cuenta que es preciso saber en cada momento dónde se van a escribir o leer los datos. Por tanto, resulta obligado fijar una posición al DMA antes de realizar una operación de intercambio con el disco. Vale cualquier posición de memoria exceptuando las comprendidas entre &4000 y &H8000, puesto que es aquí donde se pagina el controlador de disco y éste no puede leer o escribir sobre sí

Fig. 1. Disposición de datos en un FCB.

Byte	Campo	Contenido
0	X:	Drive
1	NOMBRE DEL FICHERO	
12	BLK	Bloque en curso
14	REC TAMAÑO	Tamaño de un registro
16	TAMAÑO	
20	A/M/D/	Fecha
22	H/M	Hora
24	ID	Periférico
25	DIR	Localización directorio
26	1o	Primer cluster
28	no	Ultimo cluster
30	POS	Ultimo cluster (con relación al inicio del fichero)
32	RANDOM	Registro en curso
36		

Fig.2. Disposición de datos en directorio.



mismo. Naturalmente habrá que vigilar que la zona de transferencia no "machaque" algo importante (las variables del sistema, por ejemplo).

El FCB (Fue Control Block) es la parte que se encarga de decir al DOS qué es lo que debe hacer con un fichero. Se trata, por así decirlo, de un conjunto de variables agrupadas que sirven para comunicarse con el DOS. La figura 1 recoge la composición de un FCB.

Cuando se desea abrir un fichero, borrarlo, renombrarlo, escribir en él, etc., es preciso construir un FCB, para que el DOS pueda trabajar con el fichero en cuestión.

Quizá alguien piense que el FCB es idéntico a la información que se encuentra en la zona del directorio de un disquette. Sin embargo, basta compararlo con la figura 2, que recoge la disposición de los datos de un fichero dentro del directorio, para notar la gran diferencia que existe entre ambos.

Lo que sí es idéntico a un FCB es la llamada "entrada al directorio".

Las "entradas al directorio" son una copia temporal y modificada de la información de una parte del directorio del disquette, que el DOS guarda en la RAM para agilizar el trabajo. Con ello se consigue no perder el tiempo modificando constantemente el directorio a cada cambio que se produzca en algún fichero. La información se va actualizando, en la entrada al directorio correspondiente y es escrita sólo cuando el fichero en particular se cierra o cuando se ordena un cierra en conjunto. Ello explica por qué es peligroso cambiar de disquette cuando hay un fichero abierto, ya que las entradas al directorio en memoria serán escritas sobre el nuevo disco y éste quedará inservible.

DPB Y FAT

El DPB (Drive Parametrer Block) es una tabla donde se especifican las características físicas y lógicas del disquette. Hay una por cada unidad de discos conectada y están ubicadas en la RAM. Aunque dejamos para más adelante su descripción completa, vale adelantar que la información que se guarda en el DPB va desde el número de bytes que conforman un sector hasta el cluster donde empieza el directorio pasando por punteros a la FAT.

La FAT (File Allocation Table) está grabada en los primeros sectores de un disquette y sirve para guardar una relación concreta de las partes del disco que contienen datos útiles y de las que están disponibles. Su funcionamiento, al igual que el del DPB, será tratado en el futuro.

Ahora, y a fin de entrar en materia de una forma práctica, lo mejor es empezar a trabajar con el DOS.

A continuación se describe una rutina que actúa de forma similar al conocido comando "DUMP" del CP/M o del MS/DOS.

DUMP es una de esas rutinas que se escriben en minutos y se emplean toda la vida. Su cometido, como indica su nombre, es "volcar" los datos de un fichero en la pantalla (o en la impresora). El resultado es un conjunto de líneas que muestran el contenido hexadecimal y alfanumérico de todos los bytes que hay en el fichero especificado.

LISTADO 1

```
10 '****** DUMP desde BASIC *****
20 '
30 'Debe cargarse con BLOAD"DUMP.BIN"
40 'Luego, ha de inicializarse
50 'con DEFUSR=&HC000. El DUMP se
60 'logra haciendo ?USR("NOMBRE")
70 '
80 '
90 FORX=&HC000 TO &HC12B:READ V$
100 POKEX,VAL("&H"+V$)
110 S=S+PEEK(X):NEXT
120 CLS:IF S <> 34771! THEN BEEP:PRINT"
HAY UN ERROR":END
130 PRINT"DATAS CORRECTAS";CHR$(10)
140 PRINT"Pulsa una tecla para"
150 PRINT"grabar DUMP.BIN"
160 Z$=INKEY$:IFZ$=""THEN160
170 BSAVE"DUMP.BIN",&HC000,&HC12B
180 DATA ED,73,2C,C1,CD,A6,C0,CD,DD,C0,
3A,B0,F3,D6,06,CB,3F,CB,3F,6F,26,00,22,
30,C1,CD,03,C1,CD,9F,C0,3E,0C,CD,95,C0,
6C,CD,F3,C0,22,2E,C1,3A,2F,C1,CD,84,C0,
3A,2E,C1,CD,84,C0,3E,3A,CD,95,C0,3E,20,
CD,95,C0
190 DATA 21,32,C1,ED,4B,2F,C1,C5,C5,7E,
CD,84,C0,3E,20,CD,95,C0,C1,23,10,F2,21,
32,C1,C1,C5,7E,FE,20,38,04,FE,80,38,02,
3E,2E,CD,95,C0,3E,20,C1,23,10,EB,3E,0D,
CD,95,C0,3E,0A,CD,95,C0,2A,2E,C1,ED,4B,
30,C1,09
200 DATA 18,A1,F5,1F,1F,1F,1F,CD,8D,C0,
F1,E6,0F,C6,90,27,CE,40,27,0E,02,5F,E5,
CD,7D,F3,B7,E1,C9,11,32,C1,0E,1A,18,F2,
06,0B,21,44,C1,AF,77,23,36,20,10,FB,06,
18,23,77,10,FC,13,1A,13,6F,1A,67,0E,09,
11,45,C1
210 DATA CD,D0,C0,D8,11,4D,C1,0E,03,EA,
D0,C0,23,7E,FE,2E,C8,FE,22,3F,C8,ED,A0,
E0,18,F3,11,44,C1,0E,0F,CD,98,C0,C8,11,
14,C1,0E,09,CD,98,C0,ED,7B,2C,C1,C9,E5,
21,01,00,11,44,C1,0E,27,CD,98,C0,E1,C8,
18,EB,21
220 DATA 00,00,22,65,C1,22,67,C1,3A,30,
C1,6F,22,52,C1,C9,0A,0D,46,49,43,48,45,
52,4F,20,4E,4F,20,45,4E,43,4F,4E,54,52,
41,44,4F,24
```

LISTADO 2

```
10 '****** DUMP desde MSXDOS *****
20 '
30 'Debe usarse haciendo:
40 'A>DUMP NOMBRE
70 '
80 '
90 FORX=&HC000 TO &HC0F1:READ V$
100 POKEX,VAL("&H"+V$)
110 S=S+PEEK(X):NEXT
120 CLS:IF S <> 24135! THEN BEEP:PRINT"
HAY UN ERROR":END
130 PRINT"DATAS CORRECTAS";CHR$(10)
140 PRINT"Pulsa una tecla para"
150 PRINT"grabar DUMP.COM"
160 Z$=INKEY$:IFZ$=""THEN160
170 OPEN"DUMP.COM"FOR OUTPUT AS#1
180 FOR X=&HC000 TO &HC0F1
190 PRINT#1,CHR$(PEEK(X));:NEXT
200 END
1000 DATA ED,73,F2,01,CD,A3,01,3A,B0,F3
,D6,06,CB,3F,CB,3F,6F,26,00,22,F6,01,CD
,C9,01,CD,9C,01,3E,0C,CD,92,01,6C,CD,B9
,01,22,F4,01,3A,F5,01,CD,81,01,3A,F4,01
,CD,81,01,3E,3A,CD,92,01,3E,20,CD,92,01
,21,F8,01
1010 DATA ED,4B,F5,01,C5,C5,7E,CD,81,01
,3E,20,CD,92,01,C1,23,10,F2,21,F8,01,C1
,C5,7E,FE,20,38,04,FE,80,38,02,3E,2E,CD
,92,01,3E,20,C1,23,10,EB,3E,0D,CD,92,01
,3E,0A,CD,92,01,2A,F4,01,ED,4B,F6,01,09
,18,A1,F5
1020 DATA 1F,1F,1F,1F,CD,8A,01,F1,E6,0F
,C6,90,27,CE,40,27,0E,02,5F,E5,CD,7D,F3
,B7,E1,C9,11,F8,01,0E,1A,18,F2,11,5C,00
,0E,0F,CD,95,01,C8,11,DA,01,0E,09,CD,95
,01,ED,7B,F2,01,C9,E5,21,01,00,11,5C,00
,0E,27,CD
1030 DATA 95,01,E1,C8,18,EB,21,00,00,22
,7D,00,22,7F,00,3A,F6,01,6F,22,6A,00,C9
,0A,0D,46,49,43,48,45,52,4F,20,4E,4F,20
,45,4E,43,4F,4E,54,52,41,44,4F,24
```

DESCRIPCION DE LA RUTINA

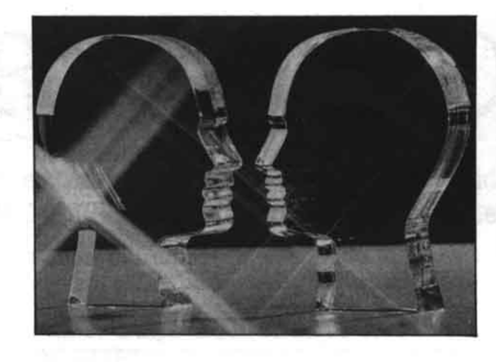
El listado de la rutina de "DUMP" no está todo lo optimizado posible. Me ha parecido oportuno separar en partes todas las pequeñas rutinas que componen el listado, en lugar de construir un programa más estructurado. El motivo no es otro que dejar claro que cada una de las partes es una pieza necesaria que se usa en esta rutina en particular y en cualquier otra que se construya para trabajar con el DOS. Lo úni-

co que puede variar es el bucle principal, que es particular a "DUMP", pero los módulos de fijación del DMA (SETDAM), ajuste del FCB (SETFCB), construcción de un FCB (MAKEFCB), etc.; son necesarios que en casi todos lios programas que corren bajo el DOS.

Dicho esto, pasemos a la descripción de la rutina.

La línea 30 guarda el puntero de pila, a fin de restaurarlo y devolver el control al BASIC si se produce un error (si no se encuentra el fichero, los datos están deteriorados o se alcanza el final de éste).

La rutina MAKEFCB (850-1180) toma la cadena de caracteres pasada desde el BASIC (apuntada por DE) y la transforma en un nombre de fichero



válido, utilizable por el DOS.

La rutina OPEN (1190-1220) abre el fichero para leerlo. El DOS pone el acumulador a cero si la apertura es exitosa. En otro caso, el sistema operativo no ha sido capaz de encontrar el fichero pedido o el nombre pasado contiene caracteres incorrectos, y la rutina volverá al BASIC mostrando el error.

Las líneas 60 a 120 leen el ancho de la pantalla (WIDTH) que se guarda en la variable del sistema situada en &HF3B0. Al final, se coloca en DAT el número de datos a representar por línea. La fórmula empleada es DAT = (WIDTH-6)/4.

La rutina SETFCB (1360-1420) pone los punteros de lectura secuencial a cero, es decir, se indican al DOS que el primer registro a leer es el cero. Asimismo, se fija a la longitud de un registro a los caracteres que contiene

L I S T A D O

```
10          ORG #C000
20 ;
30          LD   (STACK),SP
40          CALL MAKEFCB
50          CALL OPEN
60          LD   A,(#F3B0)
70          SUB  6
80          SRL  A
90          SRL  A
100         LD   L,A
110         LD   H,0
120         LD   (DAT),HL
130         CALL SETFCB
140         CALL SETDMA
150         LD   A,12
160         CALL PRINT
170         LD   L,H
180 L0:     CALL READ
190         LD   (POS),HL
200         LD   A,(POS+1)
210         CALL HEX
220         LD   A,(POS)
230         CALL HEX
240         LD   A,":"
250         CALL PRINT
260         LD   A," "
270         CALL PRINT
280         LD   HL,DMA
290         LD   BC,(DAT-1)
300         PUSH BC
310 L1:     PUSH BC
320         LD   A,(HL)
330         CALL HEX
340         LD   A," "
350         CALL PRINT
360         POP  BC
370         INC  HL
380         DJNZ L1
390         LD   HL,DMA
400         POP  BC
410 L2:     PUSH BC
420         LD   A,(HL)
430         CP   32
440         JR   C,NOVAL
450         CP   128
460         JR   C,VAL
470 NOVAL:  LD   A,"."
480 VAL:    CALL PRINT
490         LD   A," "
500         POP  BC
510         INC  HL
520         DJNZ L2
530         LD   A,13
540         CALL PRINT
550         LD   A,10
560         CALL PRINT
570         LD   HL,(POS)
580         LD   BC,(DAT)
590         ADD  HL,BC
600         JR   L0
610 ;
620 ;
630 HEX:    PUSH AF
640         RRA
650         RRA
660         RRA
670         RRA
680         CALL DIGH
690         POP  AF
700 DIGH:   AND  #F
710         ADD  A,#90
720         DAA
730         ADC  A,#40
740         DAA
750 PRINT:  LD   C,2
760         LD   E,A
770 BDOS:   PUSH HL
780         CALL #F37D
790         OR   A
800         POP  HL
810         RET
820 SETDMA: LD   DE,DMA
830         LD   C,#1A
840         JR   BDOS
850 MAKEFCB: LD  B,11
860         LD   HL,FCB
870         XOR  A
880         LD   (HL),A
890 L3:     INC  HL
900         LD   (HL)," "
910         DJNZ L3
920         LD   B,36-12
930 L4:     INC  HL
940         LD   (HL),A
950         DJNZ L4
960         INC  DE
970         LD   A,(DE)
980         INC  DE
990         LD   L,A
1000        LD   A,(DE)
1010        LD   H,A
1020        LD   C,9
1030        LD   DE,FCB+1
1040        CALL GET
1050        RET  C
1060        LD   DE,FCB+9
1070        LD   C,3
1080        JP   PE,GET
1090        INC  HL
1100 GET:   LD   A,(HL)
1110        CP   "."
1120        RET  Z
1130        CP   34
1140        CCF
1150        RET  Z
1160        LDI
1170        RET  PO
1180        JR   GET
1190 OPEN:  LD   DE,FCB
1200        LD   C,#F
1210        CALL BDOS
1220        RET  Z
1230        LD   DE,ERRO
1240 ERROR: LD   C,9
1250        CALL BDOS
1260 EXIT:  LD   SP,(STACK)
1270        RET
1280 READ:  PUSH HL
1290        LD   HL,1
1300        LD   DE,FCB
1310        LD   C,#27
1320        CALL BDOS
1330        POP  HL
1340        RET  Z
1350        JR   EXIT
1360 SETFCB: LD  HL,0
1370        LD   (FCB+33),HL
1380        LD   (FCB+35),HL
1390        LD   A,(DAT)
1400        LD   L,A
1410        LD   (FCB+14),HL
1420        RET
1430 ERRO:  DEFB 10,13
1440        DEFM "FICHERO NO ENCONTRADO$"
1450 ;
1460 ;
1470 STACK: DEFW 0
1480 POS:   DEFW 0
1490 DAT:   DEFW 0
1500 DMA:   DEFS 18
1510 FCB:   DEFS 36
```

una línea (DAT).

La rutina SETDMA (820-840) fija el área de transferencia de datos a un nuevo DMA, que es un pequeño buffer para almacenar un registro (los caracteres de una línea).

Las líneas 150-160 borran la pantalla.

La línea 170 inicializa el contador a cero.

Las líneas 180 a 600 conforman el bucle principal del programa. Se empieza leyendo un registro y se sigue imprimiendo el número de línea (190-270). Las líneas 280 a 380 imprimen el contenido hexadecimal de los caracteres de un registro, dejando un espacio en blanco entre los datos. A continuación, las líneas 390 a 520 imprimen el código de cada carácter del registro, mirando que no se trate de un código de control o de un carácter semigráfico, en cuyo caso se imprime un punto. Por último, las líneas 530 a 600 completan el bucle principal haciendo un avance de línea e incrementando el contador.

La rutina HEX (630-740) escribe los dos digitos hexadecimales del contenido del acumulador.

EMPLEO DE "DUMP"

Lo primero que hay que hacer cuando se tiene a "DUMP" en la memoria es definir la llamada con algo como: "DEFUSR = &HC000". A continuación hay que suministrarle el nombre del fichero que se desa volcar. El procedimiento a seguir se sirve de la facultad del BASIC para intercambiar parámetros con rutinas del usuario. Hay que hacer algo como: PRINT USR ("Nombre"). El intérprete colocará el nombre en la memoria y hará que el registro DE apunte al inicio de éste. La rutina, por tanto, sabrá dónde está la cadena gracias a DE.

Lo anterior está referido al uso de "DUMP" desde el BASIC, siendo imposible su empleo bajo el MXSDOS.

Ahora bien, unas pequeñas modificaciones bastan para construir un "DUMP" utilizable desde el MSXDOS.

En primer lugar hay que cambiar la dirección de origen (&HC000, en el caso del BASIC) y fijarla en &H100. El hecho de compilar la rutina a partir de la posición &H100 no es casual, puesto que se trata del inicio de la TPA (Transient Program Area). Esta zona es la que contiene los programas que corren bajo el MSXDOS (y bajo el CP/M). Cuando el MSXDOS carga un programa en la memoria lo ejecuta saltando a la posición &H100, así que todos ellos han de estar diseñados para correr a partir de esta dirección.

El segundo cambio es suprimir la rutina que toma el nombre y lo transforma en un FCB válido (MAKEFCB), puesto que el procesador de comandos (COMMAND.COM) toma el parámetro (el nombre del fichero) que acompaña al comando DUMP, construye con él un FCB utilizable y lo coloca en la posición &H5C. Huelga decir que desde el DOS los ficheros ejecutables han de llevar la extensión "COM", así que para utilizar a "DUMP" deberemos llamarlo "DUMP.COM".

Luego de efectuar las modificaciones descritas y de compilar el código en un fichero llamado "DUMP.COMP", podremos emplear la rutina haciendo:

A>DUMP Nombre

El MSXDOS comprueba si hay un identificador de unidad de disco, por tanto valen expresiones como:

A>DUMP B:Nombre

La rutina hace llamadas a funciones del DOS para escribir los caracteres en la pantalla, así que el sistema operativo comprobará constantemente si se aborta el programa, con CTRL-C, o si se hace una pausa, con CTRL-S (la tecla STOP no sirve). Hay que prestar una atención particular a no pulsar por descuido CTRL-P, porque ello conectará el eco de la impresora y bloqueará el programa hasta que ésta se halle en línea.

Resta comentar que hay dos cargadores de líneas DATA. El primero que construye "DUMP.BIN", para usar la rutina desde el BASIC, y el segundo que crea "DUMP.COM", para emplearla bajo el MSXDOS. El listado fuente de la versión MSXDOS. El listado fuente de la versión MSXDOS no aparece, por cuestiones de espacio, pero puede obtenerse teniendo en cuenta las pequeñas modificaciones ya dichas.

Nótese también que "DUMP.COM" es escrito a partir de la apertura de un fichero secuencial, para evitar que el intérprete coloque la cabecera típica de los ficheros grabados con BSAVE, y que contiene un prefijo y las direcciones de inicio, final y ejecución.

Las aplicaciones de "DUMP" pueden ser de las más variadas. Particularmente lo empleo con frecuencia, a fin de tener una idea del contenido de un fichero olvidado, estudiar los mensajes de error de un programa comercial que no hace lo que deseo (y sacar conclusiones de ellos) o mirar de qué forma un determinado programa comprime las pantallas (para poder descomprimirlas); por citar sólo dos ejemplos.

Espero que vosotros mismos descubráis otros usos de DUMP y os animéis a escribir pequeñas utilidades. Me atrevo a sugerir que construyáis un "TYPE" que no pare al encontrar un código de final de fichero (&H1A), un comando "DIR" que muestre los ficheros por orden de antigüedad, un "COPY" que emplee la RAM DE Vídeo de los MSX-2 para reducir los cambios de discos o un "COPY" que salve sólo los ficheros que han sido modificados (igual que el comando BACKUP, del MS/DOS).

BIT-BIT Software Juegos

por Ronald Van Ginkel, Sascha Ylla-Könneke, Javier Guerrero, Hirofumi Kuraoka, Willy Miragall

SPIRITS

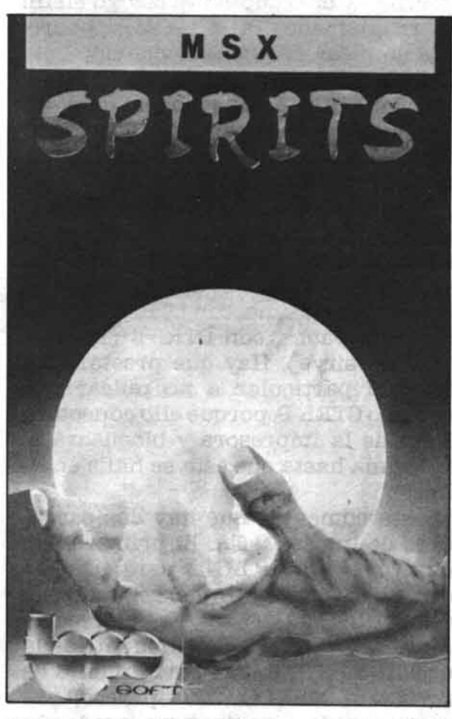
TOPO soft
Formato: Cassette, MSX-I
Controles: Teclado y joystick

Los relámpagos rasgaban la noche y la lluvia azotaba incansablemente el rostro de Wheete, el cual, vagando por las llanuras, desesperaba ya de encontrar un cobijo. Pero un relámpago estalló delante suyo, iluminando un maltrecho castillo. Le pareció distinguir unos ojos mirándole por una ventana; pero el hecho de poder resguardarse de la lluvia le llevó a llamar a la puerta del castillo. Tras una larga espera, comprobó desesperado que el castillo no estaba habitado y la puerta bien cerrada. Se dio la vuelta con el objeto de marcharse cuando oyó el ruido de una puerta al abrirse. Con paso inseguro se adentró en el castillo, débilmente iluminado. Como ya se esperaba, la puerta se cerró a sus espaldas con un golpe sordo. Temeroso, comenzó a caminar cuando sus pies toparon con un objeto ligero y plano. Se agachó y recogió el objeto, lo reconoció rápidamente, era un papel. Y con letra femenina, se leía lo siguiente: «Por favor, quien sea que haya recogido esta carta, ayúdeme. Estoy prisionera en este castillo maldito desde hace ya meses. Algunos valientes caballeros han intentado rescatarme y prefiero no describir el final que tuvieron. Existe una armadura que está hechizada y camina por los dos castillos (ignoro por cual de los dos habrá usted entrada; pero ambos castillos están comunicados). Es imprescindible deshechizar a esta armadura, con lo cual podremos debilitar al señor del castillo, que por una antigua maldición ha adoptado la forma de un águila. Odia a muerte a todos los seres humanos, de manera que usted no podrá salir del castillo mientras él viva. A mí me encontrará vagando por las habitaciones de uno de los dos castillos. Por favor, rescáteme...»

Wheete entrecerró los ojos, se ajustó la capucha de manera que su rostro permaneciese en las sombras, y conjurando a un poderoso espíritu que le debía un favor, logró cerrar un trato mediante el cual el espíritu le resucitaría cada vez que pereciese en manos de los malvados habitantes del castillo. Pero el espíritu sólo ofreció resucitarle 3 veces. Una vez hecho esto, se preparó para la aventura y comenzó a explorar la primera habitación...

Tu personaje es un misterioso encapuchado, que guarda su rostro en las sombras, excepto la nariz, que, debido a su descomunal tamaño, no se puede esconder tan fácilmente. Puede desplazarse hacia las cuatro direcciones, agacharse, saltar, disparar conjuros paralizadores y coger objetos.

Los enemigos son bastante diversos y da-



ñinos, los jorobados, fantasmas, hombres lobo y enanos son hechizables (los puedes paralizar por un tiempo), pero existen otros enemigos que bien por ser demasiado bajos (el gato) o por ser espontáneos (demonios arqueros) no pueden ser hechizados.

El contacto con nuestros enemigos nos restará energía, y cuando esta se acabe, nos restará una vida. Lanzar un hechizo también nos resta energía, de manera que hay que procurar racionarlos.

El contexto por el que transcurre el juego es un castillo, plagado de trampas (lámparas que caen o calderos hirviendo principalmente) y de puertas que no se abrirán a menos que accionemos la palanca adecuada, que puede estar bastante lejos de la puerta en cuestión. También hay escaleras para subir o bajar y también agujeros en el suelo, que como es de suponer, son sólo para bajar.

Lo que sin duda más destaca de este juego es el tratamiento que se le ha dado a la pantalla, la cual está dividida en tres zonas. La primera es donde transcurre la acción y por donde se desplaza el personaje, la segunda es la que indica el número de vidas, energía..., y la tercera indica el lugar a donde tenemos que ir para alcanzar un objeto o personaje determinado (al principio sólo nos indica el lugar donde se encuentra la bola mágica, una vez cogida ésta, podremos seleccionar con los números 1 al 5, 1- Libro, 2- Varita mágica, 3- Princesa, 4- Armadura y 5- Aguila, en el caso de estos tres últimos, la ventana seguirá a los personajes por todas las pantallas).

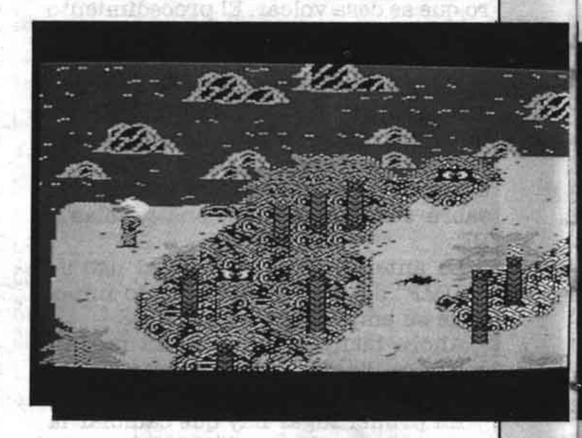
El objetivo del juego es destruir al águila, para lo cual tendremos que coger el libro, con el que deshechizaremos a la armadura. Después nos haremos con la varita para desencadenar a la princesa, y una vez hecho esto, sólo falta encontrar al águila.

Los gráficos son bastante buenos. Lamentablemente no se utilizan los sprites a la hora de representar a los personajes, provocando así que toda la zona que rodea al personaje se vuelva del mismo color que este último, lo cual desmerece bastante al programa.

PUNTUACION:
Presentación: 10
Gráficos: 7
Movimiento: 8
Música: 9
Dificultad: 9
Total: 8

Howard the Duck

Activision
Formato: Cassette MSX-I y MSX II
Controles: Joystick y Cursores
Precio: 880 Ptas.



Howard el pato estaba bailando hacía un minuto con todos sus amigos, cuando de pronto estos desaparecieron, y Howard se quedó solo. De repente Howard supo lo que pasaba, sus amigos estaban en la isla de Vulcano y el oscuro Overlord los había raptado. Inmediatamente se puso en camino, compró un billete de avión y cuando éste sobrepasaba la isla se lanzó en paracaídas dispuesto a acabar con el oscuro Overlord y rescatar a sus amigos.

Como habréis podido comprobar, el juego no tiene que ver nada con la película del mismo nombre. En el juego manejamos a un pato y tenemos que rescatar a nuestros amigos.

El juego comienza cuando ponemos los

pies en la isla, después del viaje en paracaídas. Durante la estancia en la isla Vulcano, necesitaremos un equipo completo para poder acceder a los lugares difíciles, estos son:

—Un jet solar, que te ayudará a cruzar el canal que hay en la isla.

—Un ultraligero que necesitarás para acceder a la cima del volcán de la isla.

—Un desintegrador de neutrones portátil con el que te enfrentarás al oscuro Overlord.

Lo primero que tendrás que hacer después de coger el equipo es dirigirte al canal, y situarte en la orilla, automáticamente se activará el jet solar y podrás remontar el canal. Cuando llegues a la otra orilla comenzarán a atacarte los maniacos mutantes, seres que viven bajo tierra y que tendrás que liquidar a base de patadas y puñetazos. La operación de remontar el canal y recorrer nadando la isla, se tendrá que repetir varias veces hasta que llegues a un puente tras el cual acaba la primera fase del juego. La segunda la comenzamos volando con el ultraligero hasta la boca del volcán, una vez dentro estarás en la guarida del oscuro Overlord donde tendrás que esquivar las estalactitas que caen del techo y los agujeros que aparecen en el suelo. Para colmo tendrás que procurar que no te acierten los disparos de energía que te lanzará el oscuro Overlord.

Si después de todo lo sufrido aún estás vivo, te faltará por realizar la última parte del juego, matar al oscuro Overlord con tu desintegrador de neutrones. Si estás situado en la posición correcta podrás disparar contra él, y acertándole varias veces lo matarás rescatando así a tus amigos. Pero ¡atención!, aún no ha acabado ahí la misión, desde el comienzo del juego Overlord astutamente te había dado un plazo de tiempo para intentar salvar a tus amigos, y había provocado la erupción del volcán, con su muerte no has conseguido que ésta se detenga y si no lo haces tú el volcán explotará. Para ello tendrás que dirigirte a una palanca, que moviéndola os salvará a todos de una muerte segura. Sólo así habrás concluido con éxito tu misión.

Antes de comenzar el juego, dispones de la opción de elegir entre cuatro niveles de dificultad: novato, intermedio, avanzado y experto. Si eliges el primero tu aventura acabará cuando cruces el puente y te entregarán la medalla de novato o aprendiz respecto al tiempo empleado. Si has elegido la opción segunda terminarás en la cima del volcán y te será entregada la medalla distinguida o de teniente, y los que elijan el nivel avanzado o experto y maten a Overlord les serán entregadas las medallas de comandante o capitán (avanzado) y commodore o almirante (experto). Hay que decir que en las opciones de avanzada y experta a la hora de remontar el canal es agobiante, desesperante y extraordinariamente difícil, teniendo que rendirse ante su dificultad.

Howard the Duck es un juego adictivo, animado con unos buenos gráficos, ricos en colorido y un movimiento bien conseguido. Además también está disponible para MSXII.

PUNTUACION:
Presentación: 9
Gráficos: 8
Música: 7
Adicción: 9
Movimiento: 8
Dificultad: 9
Total: 8



Fernando Martín Basket Master

DINAMIC
Formato: Cassette, MSX-I
Controles: Teclado, Cursores y joystick
Precio: 875 Ptas.

Hacía mucho tiempo que los usuarios de ordenadores MSX pedían a gritos algún juego dedicado a este espectacular deporte como es el baloncesto. Dinamic fue, al fin, la que se decidió a realizarlo, y lo hizo de una manera bastante original, como es llevando a nuestras pantallas a un jugador tan conocido como Fernando Martín. En el juego no tendréis que enfrentaros un equipo contra otro, sino que jugaréis tú y él solos, esto sin duda beneficia la maniobrabilidad del jugador, y elimina los posibles líos de control con que te podrías encontrar si jugara un equipo completo contra otro. La pantalla está ocupada por todo el campo de juego, cada jugador, como es obvio, defiende su canasta e intenta encestar en la contraria. Y como si de un partido real se tratara podrás lanzar tiros de tres puntos, robarle balones a tu contrincante, tirar normal o de gancho, realizar tapones, hacer dobles, fueras, campo atrás y conseguir ocho tiros de mate que se ejecutan de modo automático según nuestra posición bajo la canasta, estos son:

—De frente a dos manos.

—De espaldas con las dos manos.

—De lado con la mano derecha a estilo gancho.

—De lado con la mano izquierda a estilo gancho.

—De frente a dos manos y giro en el aire.

—De espaldas a dos manos y giro.

—De lado con mano derecha a estilo gancho y giro en el aire.

—De lado con la mano izquierda a estilo gancho y giro en el aire.

Bajo el campo de juego están situados los marcadores individuales que nos indican el estado de fuerza de los jugadores, el número de untos que ha conseguido cada uno, y la falta que ha cometido (personal, dobles...). Hay otro indicador que marca el tiempo que resta para la finalización del partido. En cuanto a los gráficos de este juego cabe decir que la cancha y las canastas están bien realizadas, pero los espectadores de las gradas son bastante chapuceros. Aparte de esto, los programadores no se han molestado mucho con dar a los jugadores un aspecto deportivo y real, ya que les han plantado una cabeza demasiado grande y un cuerpo muy corto y comprimido. Tampoco se han molestado en darles una silueta diferente, ya que aparte de la cabeza el resto del cuerpo es el mismo en los jugadores.

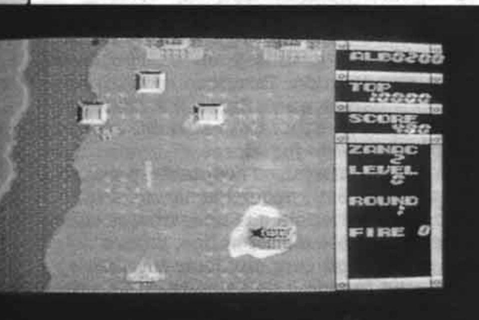
El juego en general, es aceptable, pero excesivamente difícil, ya que Fernando Martín tiene más experiencia que nosotros, se cansa menos, comete menos fallos, y para colmo es muy difícil (por no decir imposible) hacerle una jugada y dejarlo atrás. No obstante existe un truco que nos permitirá meter tantas canastas como queramos, sin ningún problema. Cuando él tenga la pelota dirigirnos a su canasta y saltar, entonces la pelota aparecerá en nuestras manos y se introducirá directamente en la canasta sumándonos dos puntos.

Hay que elogiar al gráfico de presentación que surge al cargar el juego, es magnífico. Aparece Fernando Martín apoyado en una pierna, dando vueltas con un dedo a la pelota y ofreciéndonos una de sus mejores sonrisas.

PUNTUACION:
Presentación: 9
Gráficos: 7
Gráficos: 7
Música: 9
Adicción: 6
Movimiento: 6
Dificultad: 9
Total: 8







ZANAC

PONYCA
Formato: Cassette MSX 1/2
Controles: Joystick/teclado
Precio:

He aquí un juego muy adictivo, en el que la música contribuye positivamente a esta «adicción».

A este juego se le puede atribuir bien el nombre de «matamarcianos», ya que nuestra misión es la de matar marcianos, ojos y otros muchos seres y naves que intentarán por todos los medios restarnos una y otra vez vidas para así tener que dejar de jugar.

En este juego hay que destacar el disparo de nuestra nave, ya que lo podemos mejorar recogiendo unas bolitas. También existen más ayudas, lo que viene siendo habitual en los nuevos juegos, como por ejemplo vidas extra, pasadizos secretos a los siguientes «rounds», ayuda en el disparo (aparte de la ya mencionada) y varios escudos diferentes.

Como en todos los juegos de arcade aquí también tenemos que ir sumando puntos y pasar obstáculos. Los obstáculos son muy diversos aunque la mayoría son meteoritos, naves enemigas (de todas las formas y colores), cohetes, marcianos en general y unos ojos a los que odiaréis ya que sólo si los matáis podréis seguir jugando —sobrevolando las tierras enemigas— lo que sin duda os costará mucho.

Lamentablemente los programadores no quisieron hacer casi perfecto este juego ya que la parte gráfica —los lugares que sobrevolamos— es muy pobre y repetitiva. Siendo la presentación, la música en sí y el resto del juego (con la excepción de los gráficos) muy buenos, tenemos que mencionar que se trata de un Arcade muy difícil en el que hay que dominar y saber todos los trucos y ayudas que nos ofrece el juego, para conseguir una puntuación buena.

Este juego sin duda pondrá a prueba los reflejos de los más hábiles, haciéndoles pasar así buenos o malos ratos...

PUNTUACION:
Presentación: 10
Gráficos: 5
Música: 9
Adicción: 9
Movimiento: 8
Dificultad: 9
Total: 8

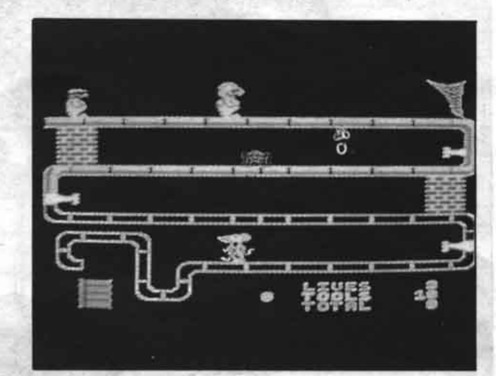
SPLASH

MIND GAMES ESPAÑA
Formato: Cartucho y cassette 64K MSX I
Controles: Cursores y joystick
Precio:

Parece ser que los juegos tipo Arcade se están volviendo a poner de moda y no es para menos, ya que son los que logran que pasemos más horas delante de nuestra máquina, debido a sus altos índices de adicción y entretenimiento. Este es el caso de Splash, un Arcade típico que no llamaría la atención si no fuera por su gran originalidad.

La acción se desarrolla en una instalación de tuberías, separadas por unos grifos cerrados impidiendo el paso del agua. Como es natural, nuestra misión consistirá en abrirlos todos y, de esta forma, el agua podrá circular tranquilamente por las tuberías hasta desembocar en un gran barril cuya capacidad es de mil litros. Cuando este barril esté lleno, habremos conseguido pasar la primera pantalla y tendremos que hacer lo mismo en la siguiente, en la otra, etc... Pero como no sabemos nada de grifos, necesitaremos la ayuda de un profesional de la materia, el fontanero, al cual tendremos que guiar por las tuberías hasta los grifos cerrados y éste nos los abrirá.

Hasta aquí todo parece fácil. Quizás el único problema radica en guiar al fontanero que es algo despistado y se suele quedar atrás. Pero dos peligrosos seres se arrastran por las instalaciones y al menor descuido que tengamos nos empujarán y nos harán caer a trompicones por las tuberías, estos son: una araña y un ratón. El más fácil de esquivar es la araña, de movimientos lentos y pesados; pero eludir al ratón es dificilísimo. Sus movimientos son rápidos e inesperados no dándote tiempo a reaccionar. La única forma de darte un respiro es tirarle a la cabeza una llave inglesa, y aunque no lograremos eliminarle sí conseguiremos aturdirlo durante un rato, tiempo suficiente para abrir algún grifo.



Pero el pícaro ratón no sólo estorbará nuestro camino e intentará empujarnos, sino que también cerrará los grifos abiertos, teniendo que doblar así nuestro esfuerzo.

El juego en general está bien realizado, sobre todo por su originalidad y adicción. Está animado por unos gráficos apropiados para su estilo y unos sprites grandes y bien conseguidos. En definitiva, un juego nuevo que gustará a los pequeños de la casa y sorprenderá a los mayores.

PUNTUACION:
Presentación: 5
Gráficos: 7
Movimiento: 7
Música: 8
Adicción: 9
Dificultad: 8
Total: 7

GLASS

MIND GAMES
Formato: Cassette, MSX-I
Controles: Cursor y joystick

«Se busca tripulante experimentado para formar parte de la tripulación de la nave espacial ANDROMEDA V. Aspirantes preséntensen en las oficinas de VIAJES A GO GO, S.A.»

¿Te apetece formar parte de esta nave espacial? Si es así, intenta pasar las pruebas de admisión con la mejor puntuación posible para lograr un buen puesto en la nave y asegurarte un buen sueldo. Pero ten cuidado, los aspirantes son muchos y las pruebas



exigen nervios de acero y una mente que tome decisiones en milésimas de segundo. Estas son las pruebas:

1— Destruye a las naves que intentarán colisionar con tu scanner continuamente.

2— Desplaza tu scanner procurando destruir a todos los enemigos que te aparecerán por los lados, esquivando a la vez las balas que ellos te disparen.

3— Destruye a las naves que irán apareciendo poco a poco por el horizonte, tienes que hacerlo rápidamente, antes de que te disparen.

4— Esta fase es de Bonus, controlas a un pequeño cuadradito, con el cual tienes que apuntar a los radares y a los trenes de aterrizaje de las naves que aparecen por la derecha de la pantalla.

5— En esta prueba debes controlar tu nave intentando esquivar los obstáculos que aparecen por pantalla.

Estas pruebas no deben aparecen necesariamente en este orden, sino que a medida que vayas pasándolas, aparecerá una que ya hayas pasado; pero con distintos gráficos o mayor dificultad.

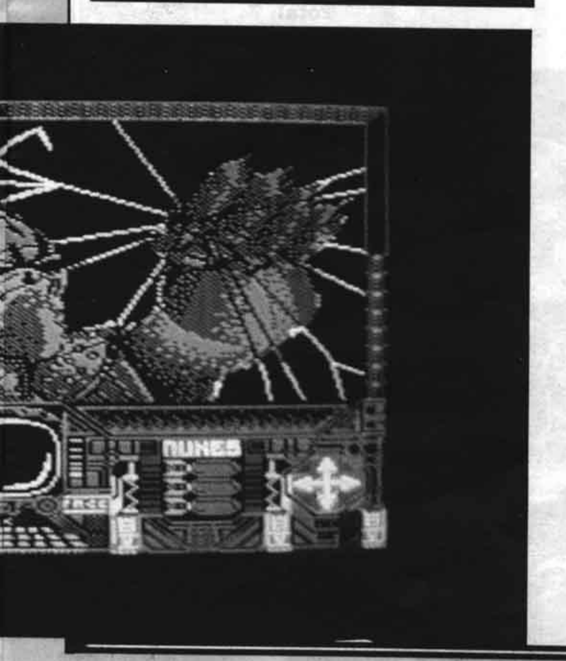
La pantalla representa el tablero de mandos de una nave, en el cual están representados el tiempo que te resta en cada prueba, la energía que te queda y el número de vidas. En la parte de arriba hay una ventana en la cual aparecerán los enemigos, obstáculos, etc.

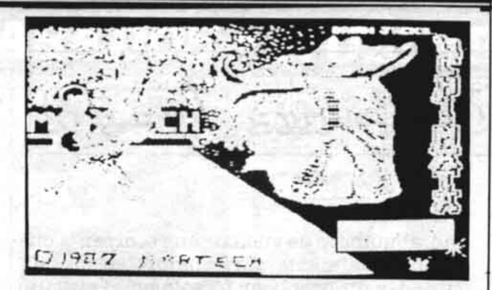
Los controles son derecha e izquierda para mover la nave a uno u otro lado, y disparo para disparar el láser, que saldrá en forma de dos rayos que se paran en el centro de la pantalla, lugar donde habrá que situar a los enemigos antes de disparar.

La nave se desplaza por una especie de pista de hielo, sobre la cual aparecerán los enemigos, reflejándose en el hielo.

En la pantalla de obstáculos, hay que moverse intentando no chocar de frente con un obstáculo, ya que cada choque resta una cierta cantidad de energía, y en un choque de frente se pierde casi una vida.

PUNTUACION:
Presentación: 8
Gráficos: 8
Movimiento: 6
Adicción: 6
Música: 6
Dificultad: 8
Total: 7





UCHI-MATA

MARTECH
Formato: Cassette 64K. MSX I
Controles: Cursores y Joystick
Precio:

Uchi-Mata es otro de los muchos simuladores de artes marciales que están disponibles en el mercado MSX. Lo que diferencia a éste de los restantes es que es el primero que está dedicado al judo, antiguo arte japonés de combate sin armas. En la mayoría de los otros juegos dedicados a las artes marciales, el protagonista o el contrincante se defendía o atacaba con algún tipo de arma, como palos, espadas, cadenas, bolas de fuego..., pero lo que diferencia a éste es que las peleas son cuerpo a cuerpo, no podemos dar patadas ni puñetazos, nuestra misión consiste únicamente en agarrar y tirar al oponente.

Este juego, como muchos otros, nos permite las posibilidades de pelear contra otro jugador, contra el ordenador y también hacer prácticas contra un oponente que no responde a nuestros ataques. Esta última opción es muy importante, ya que nos permite entrenar las llaves para la pelea, que son de muy complicados movimientos.

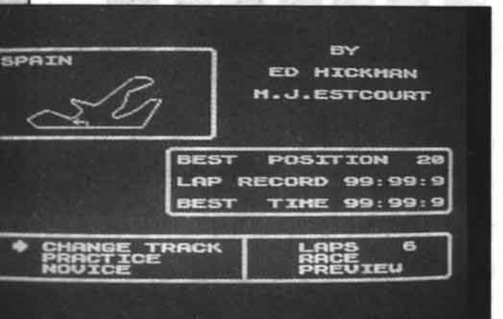
Conforme vayamos derrotando a nuestros contrincantes irán apareciendo otros con mayor experiencia que intentarán eliminarnos. Durante el desarrollo de la pelea estaremos bajo los atentos ojos de un juez que dará su puntuación a cada jugador después de que halla caído al suelo alguno de ellos. Además también se nos mostrará la posición de los pies en el momento en que se vaya a hacer la llave, y el aguante o fuerza de cada jugador.

El juego ha sido diseñado bajo el asesoramiento de uno de los mejores luchadores de judo del mundo. Es un juego muy completo y da una buena sensación de realidad. Acompañado de unos gráficos apropiados y un movimiento bastante real pero mejorable, «Uchi-Mata» es un juego original y aceptable, que a buen seguro os gustará a todos, después de haber superado su difícil manejo.

Cabe mencionar la cantidad de llaves que puedes hacer a tu oponente durante el combate, más de ocho en todas direcciones. El combate lo vencerá el jugador que reúna el mayor número de puntos, pero también puede ser ganado antes de que terminen los dos minutos de tiempo, por medio de una llave mortífera que te lanzará por los aires y te dejará K.O. El nombre de esta llave es el mismo que el del juego, «Uchi-Mata».

PUNTUACION:
Presentación: 8
Gráficos: 7
Sonidos: 7
Adicción: 8
Movimiento: 6
Dificultad: 8
Total: 7





SPEED KING

Mastertronic
Formato: Cassette, MSX-I
Controles: Cursores y joystick
Precio: 499 Ptas.

Speed King es un juego en el que tendremos que demostrar nuestras cualidades como piloto de motos de gran cilindrada. Recorreremos los mejores circuitos del mundo y competiremos procurando conseguir la mejor posición y buscando hacer la vuelta más rápida al circuito.

En la parrilla de salida habrán veinte motos delante tuyo, a las que tendrás que adelantar procurando no chocar con ellas ni salirte de la calzada, ya que acabarás por los suelos y perderás a los demás pilotos (los cuales nunca se caen ni tropiezan los unos con los otros). Pero tu moto es más rápida que las demás, consta de seis velocidades y puede alcanzar en recta hasta los 250 km/h, con lo que si te arriesgas tomando las curvas en quinta o sexta marcha podrás alcanzar a los restantes pilotos sin muchos problemas. Pero hay que tener en cuenta que existen dos tipos de curvas, las normales que las puedes pesar a altas velocidades y las cerradas y peligrosas, en las que se tiene que reducir considerablemente la velocidad y tomarlas con cuidado. Pero si además os encontráis con una de estas curvas en medio de la cadena de motos, es bastante probable que terminéis por los suelos.

El juego en general es bastante completo, pudiendo elegir muy variadas opciones. Podrás elegir entre diez famosos circuitos del mundo (San Marino, Italia, Silverstone, Paul Ricard...) entre tres niveles de dificultad, el número de vueltas, en recorrer el circuito visualmente para comprobar su dificultad, y en practicar tú solo en el circuito que elijas.

El juego es bastante pobre en gráficos, ya que no se ve más que el circuito y ningún decorado a los alrededores. Por otra parte los gráficos de las motos y los bordes de la carretera no están hechos punto por punto, sino que más bien parecen hechos en screen 3. El movimiento en general está bien conseguido dando una buena sensación de realidad.

Cabe decir que es un juego bastante adictivo y original, el primero que trata el tema de las motos para MSX, pero que si los programadores se hubieran molestado en mejorar los gráficos sin duda hubiera sido un excelente juego.

PUNTUACION:
Presentación: 5
Gráficos: 5
Música: 7
Adicción: 8
Movimiento: 7
Dificultad: 8
Total: 7

WINTER GAMES

EPYX
Controles: Teclado o joystick
Formato: Cassette MSX1
Precio:

Seguro que casi todos los amantes de los videojuegos han oído hablar de este soft de «juegos olímpicos de invierno». Ahora nos llega la versión para MSX, pero lamentablemente esta versión deja algo que desear si se la compara con la de otras máquinas.

En primer lugar echamos en falta el poder elegir el país al que representas en estas competiciones y con ello también faltan los himnos nacionales. Después de no haber podido elegir tu país, sí puedes elegir entre estas diferentes opciones:

1— compete in all events
2— compete in some events
3— compete in one event
4— practice one event
5— select number of players
6— change game controls
7— see world records

Tras elegir la opción de competir tendremos que introducir nuestro nombre.

Otra cosa que hay que mencionar es que en esta adaptación faltan muchos «events», con lo que el juego pierde bastantes alicientes cosa que no ocurre en la versión de Commodore y se hace muy corto y hasta monótono si se juega mucho.

Para los usuarios de MSX quedan estas competiciones:

1— Hot dog
2— Speed skating
3— Ski jump

con lo que faltan:

— Figure Skating
— Free Skating
— Blathalon
— Bobsled

Después de cada «event» nos es mostrado el ganador y la calificación de los otros jugadores, recibiendo los 3 primeros una medalla. Las medallas dan puntos y el que tiene al final más puntos es el vencedor de estos emocionantes juegos de invierno.

Quiero remarcar otra vez la poco adecuada adaptación para los MSX aunque esto puede ser debido a que el juego no se va cargando event por event, sino, al contrario, de un golpe. (Lo primero no habría sido problema y no defiende la versión...)

Otro defecto en la versión MSX es que en las instrucciones vienen descritas todas las competiciones o pruebas aunque, como ya he mencionado, no se puedan jugar todas.

Si queréis oír unas pequeñas melodías sólo tenéis que esperar un poco antes de competir, o sea cuando salga la pantalla del event esperar unos segundos.

PUNTUACION:
Presentación: 4
Gráficos: 9
Música: 7
Adicción: 8
Movimiento: 8
Dificultad: 7
Total: 7



NUMEROS ATRASADOS • NUMEROS ATRASADOS


MSX 2.ª Edición
N.os 1,2,3,4 - 450 PTAS.


MSX 2.ª Edición
N.os 5,6,7,8 - 475 PTAS.


MSX 2.ª Edición
N.º 9, 10, 11, 12, 13 - 575 PTAS


MSX 2.ª EDICION
N.º 14, 15, 16, 17 475 PTAS.



MSX18 175 PTAS.


MSX19.20 350 PTAS.


MSX21 175 PTAS.



MSX CODIGO
MAQUINA - 275 PTAS


MSX22 175 PTAS.


MSX23 175 PTAS.


MSX 24 175 PTAS.


MSX25.26 350 PTAS.


MSX27 225 PTAS.


MSX 28 225 PTAS.


MSX 29 225 PTAS.

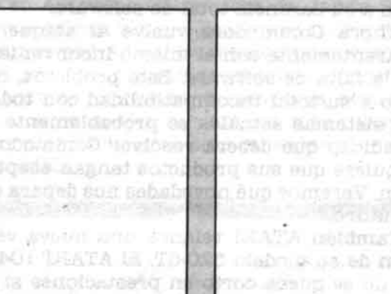

MSX 30 225 PTAS.

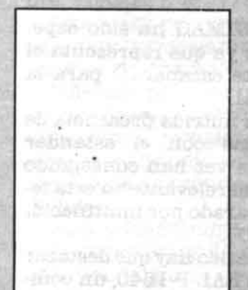

MSX 31 225 PTAS.


MSX 32, 33 450 PTAS.


MSX 34, 225 PTAS.


MSX 35 275 PTAS.





¡LA 1.ª REVISTA DE MSX DE ESPAÑA!

PARA QUE NO TE QUEDES CON LA COLECCION INCOMPLETA SOLO TIENES QUE ENVIAR HOY MISMO EL BOLETIN DE PEDIDO CON TUS DATOS PERSONALES A «SUPER JUEGOS EXTRA MSX» –DPTO. SUSCRIPCIONES C/. Roca i Batlle, 10-12, 08023 Barcelona.

BOLETIN DE PEDIDO

Deseo recibir los números .... de SUPERJUEGOS EXTRA MSX

para lo cual adjunto talón del Banco .... n.º .... a la orden de Manhattan Transfer, S.A.

Nombre y apellidos ....

Dirección .... Tel.: ....

Población .... DP. .... Prov. ....

«No se admite contrarreembolso»

SONIMAG 87: LA GRAN FERIA DE LA IMAGEN Y EL SONIDO



Pese a que aparentemente tengan poca relación con nuestra publicación, queremos resaltar las principales novedades que encontramos en esta edición de la Feria Internacional del Sonido y de la Imagen (SONIMAG).

Esta edición de SONIMAG ha sido especialmente interesante ya que representa el 25 aniversario de este escaparate para la electrónica más actual.

Hay que destacar la nutrida presencia de empresas relacionadas con el estándar MSX, que por primera vez han conseguido que éste tenga un papel relevante en esta feria (hasta ahora acaparado por multinacionales como AMSTRAD).

En el campo informático hay que destacar la aparición del AMSTRAD P-1640, un compatible PC que incorpora, como opción más destacable unos gráficos en color de tipo EGA. EGA (Enhanced Graphics Adapter) es un dispositivo que permite que los compatibles PC puedan funcionar con una resolución gráfica similar a la de los MSX de segunda generación.

Esta característica hace que este ordenador se convierte en uno de los más deseables en el difícil mercado PC. Su precio, aun en la línea AMSTRAD de bajo coste, se eleva por encima de las 200.000 pesetas.

Por este y otros motivos no constituirá un rival importante para los MSX de segunda generación, que además de su excelente resolución incorporan memoria de VRAM separada y comandos de diseño y SCROLL por hardware, aparte de la utilización de sprites. Queremos remarcar, por tanto, que pese a su excelente calidad de imagen, no deben temer nada los usuarios de MSX-2.

También dentro del apartado informático de la feria, encontramos dos interesantes novedades, que en realidad no son tales.

La primera de ellas es la aparición de dos nuevos modelos de la conocida firma Commodore. Se trata de AMIGA 500 y AMIGA 2000. Estos ordenadores son, sin duda alguna, los mejores ordenadores domésticos existentes en la actualidad. Commodore, hace ya un par de años, lanzó al mercado un revolucionario ordenador, AMIGA, que no tuvo prácticamente ninguna aceptación debido a su carencia total de software.

Ahora Commodore vuelve al ataque, y aparentemente con el mismo inconveniente, la falta de software. Este problema, debido a su total incompatibilidad con todos los sistemas actuales es probablemente el handicap que deberá resolver Commodore si quiere que sus productos tengan aceptación. Veremos qué novedades nos depara en el futuro.

También ATARI relanza una nueva versión de su modelo 520-ST. El ATARI 1040-ST no se queda corto en prestaciones si lo comparamos con AMIGA (aunque sea algo inferior en velocidad y prestaciones); pero tiene su mismo inconveniente, la total incompatibilidad y el escaso software desarrollado. Como en el caso de AMIGA, y su anterior modelo, ATARI 520-ST pasó prácticamente desapercibido en nuestro mercado.

Saliendo del mundo informático encontramos otras novedades de interés en esta edición de SONIMAG. Tal vez la más espectacular sea la presentación, por parte de SONY, del DAT (Digital Audio Tape) que consigue conjuntar la tecnología digital del Compact Disc con la posibilidad de grabación y reproducción, por tratarse de una cinta magnética.

Este producto empezará a comercializarse en breve en Alemania, y su implantación en España se retrasará hasta observar los resultados obtenidos en ese país europeo. Por el momento su precio será algo elevado, calculándose éste alrededor de las 250.000 ptas.

También por parte de SONY una implantación decisiva, ya que por primera vez se comercializa en España un sistema integrado que compagina el VIDEO-DISCO, y la más avanzada tecnología informática. Este sistema integrado permite la realización de sesiones interactivas con el VIDEO-DISCO, obteniéndose una respuesta instantánea con sólo pulsar una zona de la pantalla. Gracias a esta tecnología, se consigue que un ordenador dé una información totalmente personalizada, conjuntando imagen de vídeo con imagen de ordenador.

Por último comentar la aparición del vídeo de alta definición, que pese a estar expuesto en la feria, no se comercializará por el momento en nuestro país debido a la utilización del sistema PAL, al cual no está todavía acondicionado dicho aparato.

Estas son las novedades más espectaculares; pero SONIMAG-87 se ha caracterizado por la presentación de un verdadero torrente de éstas, en todos los campos: vídeo profesional, música, informática, etc.



MCP-4026 EL MONITOR IDEAL

Elbe, mediante su sección de microsistemas, presentó recientemente un nuevo y excelente monitor compatible con todo tipo de computador personal con salida compuesta o sistema de videocassette y video disco. Su conexionado no puede ser más sencillo, mediante conector tipo jack. Opcionalmente, el monitor modelo MCP 4026 puede equipar un altavoz de audio. Integra un tubo de rayos catódicos tipo Tv de 14 pulgadas, con un ángulo de deflexión de 90 grados, siendo el diámetro del cuello de 29,1 mm.

Dispone de entrada de señal de vídeo compuesta sistema PAL. Entrada de señal de los datos de 1,0 Vpp. La impedancia de entrada de la señal de vídeo es la usual de 15 ohmios. El barrido es de 15,625 Khz de frecuencia horizontal y de 50 Hz de frecuencia vertical.

EN PANTALLA



Posee controles exteriores de volumen y brillo en el frontis, interruptor, color, contraste, centrado vertical, horizontal y conmutador selector de señales de entrada en parte trasera.

Otras características técnicas son:

Vídeo, amplificación lineal, resolución horizontal en el centro de 250 líneas. Area de Visualización recomendable de 240 mm/145 mm. Caracteres de 8×8 puntos.

Potencia de salida de audio: 1,2 W.

Entrada de red: 220 V CA ± 10 %.

Consumo: 75 W

Condiciones de funcionamiento: temperatura, desde 0 a 40 grados centígrados.

Humedad desde el 20 % hasta el 80 %.

Para más información dirigirse a ELBE Microsystems. C/ Moyanes, 19-27, 08014 Barcelona. Tel: 421 11 11.

ASCII

ASCII CORPORATION

FRANK BERBERICH AFIRMA: «No habrá MSX 3»

El pasado día 29 de septiembre, durante la vigesimoquinta edición de Sonimag nuestra Directora Ejecutiva, Birgitta Sandberg junto con Javier Guerrero y Willy Miragall celebraron una amigable comida con Frank Berberich, Director de la Sección Internacional de ASCII CORPORATION/División MSX, en el transcurso de la cual se abordaron temas que seguro que interesarán a nuestros lectores.

En primer lugar, interrogado acerca de la noticia que ha publicado INPUT MSX sobre los MSX 3, en la cual se afirmaba que estos nuevos MSX estarían basados en el microprocesador INTEL 80386 (que incorporan los nuevos equipos de IBM) se mostró más que asombrado con semejante que desmintió categóricamente.

Es decir, es absolutamente falso que la próxima generación de MSX esté basada en el microprocesador anunciado por INPUT, lo cual demuestra su exuberante capacidad imaginativa y el poco rigor de sus fuentes de información. Señores de INPUT: Los bulos a todo color también son bulos.

La próxima generación de MSX, que está en la última fase de su desarrollo, no se llamará MSX 3. En esta nueva generación existirán simultáneamente máquinas muy diversas, como serán el MSX-E, destinado a la enseñanza; el MSX-G, pensado para los juegos, etc. De todos modos, tan pronto como recibamos información al respecto de ASCII nuestros lectores serán los primeros en enterarse (como siempre).

Así pues estos nuevos MSX estarán basados —al igual que sus antecesores— en el Z80, aunque se añadirán coprocesadores específicos para gráficos, sonido, etc. con lo que se optimizarán el rendimiento y la velocidad.

Mr. Berberich se mostró muy interesado en mantenerse en contacto permanente con nuestras publicaciones, las únicas que merecen su entera confianza que circulan regularmente en ASCII Japón, e incluso se mostró interesado en colaborar en ellas, por lo que desde aquí anunciamos que pronto tendremos su autorizadísima firma en nuestras páginas.

Otro avance acerca de MSX es que sus equipos ya se están utilizando regularmente en entornos telemáticos en Japón y en otros países, y que —Telefónica mediante— pronto podrán utilizarse análogamente en España. Ampliaremos informaciones en próximas ediciones de nuestras revistas, que como podéis ver son las más internacionales de nuestro país referidas al estándar MSX.

NUEVO SOFT

Coincidiendo con la celebración de SONIMAG 87, la conocida firma PROEIN ha lanzado tres nuevos títulos al mercado MSX.

En esta ocasión, y a diferencia de lo que ha ocurrido con otros juegos, se trata de versiones originales para MSX, es decir, no se trata de adaptaciones de juegos de otros ordenadores, por lo que aprovechan en gran medida todas las posibilidades que ofrecen nuestros ordenadores.

Los títulos de estos nuevos juegos son «DEMONIA», «MAZE MAX» y «RUNNER».

MAZE MAX es un interesante juego de conducción, en el que debes recoger un gran número de banderas que encontrarás en tu camino. Una lucha por recorrer kilómetros con el menor combustible y esquivando a todos los oponentes, que intentarán dejarnos fuera de combate en todo momento.

RUNNER, en cambio, es un juego de aventuras, en que debes recoger diferentes objetos y ofrecérselos a unos personajes determinados para conseguir llenar al final del programa.

Finalmente, DEMONIA, es un terrorífico juego, en el que debes enfrentarte a las fuerzas del mal que ve invadidos sus dominios y amenazado su poder.

LO ULTIMO DE HAL





Lo último de HAL en el mercado MSX es un cartucho que se comercializa a partir de ahora en España. Su nombre, PLANETA MOVIL.

Este juego, con unos excelentes gráficos y una muy lograda animación, nos sitúa en el interior de una nave estelar.

Debemos recorrer una interminable serie de corredores, destruyendo a toda clase de alienígenas, recogiendo algún que otro objeto, y localizando ascensores y puertas secretas que nos conducirán a nuevos niveles de juego.

Para defendernos de todos estos enemigos sólo disponemos de nuestra arma láser. Y atención, algunos de ellos son sumamente inteligentes, y no se dejan disparar as como as. Un interesante juego para ampliar vuestra programoteca con un nuevo título.

Pero en estos últimos tiempos HAL ha trabajado duro, y nos sorprende con otros dos excelentes títulos. El primero de ellos es DUNK-SHOT, uno de los mejores juegos de baloncesto en MSX que, a diferencia de otros muchos títulos, incluye el control de varios jugadores en cada equipo.

Otro título de interés para los usuarios del estándar es EGGERLAND, un juego aparentemente simple; pero que se complica extremadamente en niveles superiores. Es una perfecta combinación entre acción e inteligencia. Os lo recomendamos si os gusta pensar en los juegos.

Por último decir que estos y otros juegos de HAL son distribuidos por ZAZA Software.

ZAZA Software
C/ Loreto, 2
Tel.: 322 7 03
417 69 75
08029 BARCELONA

EN PANTALLA

ASCII Corporation en España

Como ya habíamos anunciado, Mr. Frank Berberich, director de la sección internacional de ASCII Corporation, realizó una visita a nuestro país para estudiar y potenciar la situación actual de los MSX en España. Esta visita, así como otros contactos de gran interés, son el fruto de nuestra política de relaciones internacionales que nos están permitiendo acceder a novedades en exclusiva, e informar a los importadores españoles de los productos existentes en otros mercados y del nivel de aceptación que estos tienen. Prueba de ello son, por ejemplo, SUPER RAMBO ESPECIAL (ya comercializado en España) y que comentamos en absoluta primicia, TURBO 5000, y otros programas y periféricos de los que hablaremos más adelante.

ASCII Corporation, como ya sabrán muchos de nuestros lectores, es la compañía creadora del estándar MSX. Todas las decisiones de alto nivel parten de ella. Una de las más importantes en estos momentos es el desarrollo de una nueva generación de MSX.

Mr. Frank Berberih visitó nuestras instalaciones en SONIMAG-87, teniendo con nosotros una intensa jornada de trabajo, en que comentamos entre otros muchos temas el futuro de los MSX, que mantendrán la compatibilidad por encima de todo, los nuevos periféricos y programas que están siendo desarrollados en Japón, y la práctica seguridad de que los MSX serán los ordenadores más destacados en la olimpiada de Seúl-88, ya que serán instalados como terminales inteligentes de TELE-TEXTO, informando a atletas, periodistas, etc. de todos los incidentes y datos relativos a tal acontecimiento deportivo. Esto último demuestra la enorme implantación ue los MSX están teniendo en el campo de las comunicaciones en el resto de países del mundo.

Y aún hay más, mucho más... En próximos números desarrollaremos con más detalles todos los temas tratados con Mr. Berberich, que nos confirmaron el esperanzador futuro que los MSX tienen que jugar, en nuestro país, y en el resto del mundo.

SPEEDKING AUTOFIRE

SERMA presenta una nueva versión del conocido joystick de KONIX, SPEEDKING. La primera versión, caracterizada por su gran ergonomía (comodidad y facilidad de uso), no se deja de fabricar, sino que, al contrario, se suministrará a partir de ahora con un videojuego de regalo.



Ya lo sabéis. Por el precio de SPEEDKING conseguiréis también un juego de actualidad, incluido en la caja con el joystick.

Pero KONIX también ha querido crear algo nuevo, el SPEEDKING AUTOFIRE. Gracias a esto será más fácil derrotar a las ingentes avalanchas de alienígenas que periódicamente invaden las pantallas de nuestros MSX. Como muchos ya sabréis, el AUTOFIRE consiste en un interruptor que permite la repetición automática del disparo, con lo que podemos concentrarnos únicamente en esquivar a los diferentes enemigos y dificultades que encontraremos a lo largo de nuestros juegos preferidos.

MAZE OF GALIUS

Nuevo MEGA-ROM de Konami

Un castillo mágico que, en realidad, no era sino una secreta trampa tendida por el máximo mandatario del mal, Galius.

Afrodita fue capturada por los oscuros Hunos y tú debes rescatarla. Debes tener en cuenta los terribles peligros que te esperan dentro del castillo, una trampa tendida esperando tu aparición. Nadie consiguió salir





nunca de ella; pero tú puedes intentarlo.

KONAMI, de la mano de SERMA, nos presenta su último MEGA-ROM para los MSX: The Maze of Galius, que, a diferencia de otros títulos, está preparado para funcionar en cualquier ordenador del estándar, y no únicamente en los MSX de segunda generación. Esperamos poder comentar en breve este interesante programa.

SERMA también nos presenta otro de los últimos desarrollos de KONAMI. Se trata de Q-BERT, el popular juego de ATARI, ahora desarrollado on unos excelentes gráficos para los MSX.

Q-BERT debe saltar sobre los cubos de colores y hacerlos girar. Si consigues alinear cinto cubos en línea, ¡premio! Y aunque la cosa parece fácil...

Además permite que participen dos jugadores simultáneamente.

ALBUM DE PLATINO

Sotfware excepcional a un precio excepcional

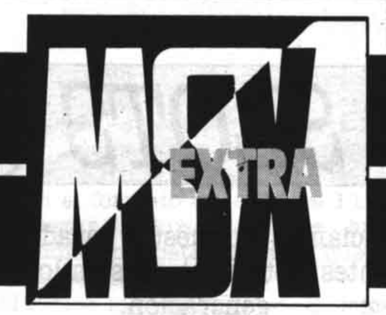
SERMA, distribuidora en exclusiva de los productos de KONAMI en nuestro país, lanzó hace ya unos cuantos meses una serie de programas en cinta enmarcados dentro de la serie CODE MASTERS.

Algunos de estos juegos ya han sido comentados en nuestra sección BIT-BIT, teniendo una interesante calidad todos los comentados hasta el momento.

En vista del interés que ha generado esta serie de programas de bajo precio, SERMA acaba de anunciar el lanzamiento de su primer ALBUM DE PLATINO.

El album de platino no es más que una recopilación de cuatro títulos de la serie CODE MASTERS, cada uno con su propia cinta, en un estuche plateado que permite la conservación de las cintas en el mismo.

Pero lo más interesante de este primer album de platino es su precio, ya que las cuatro cintas, con el estuche, costarán aproximadamente 2.000 ptas., precio que sitúa estas cintas dentro de las más asequibles que pueden encontrarse actualmente en nuestro mercado.

4.º GRAN  CONCURSO DE PROGRAMAS

COMO DE COSTUMBRE... ¡PREMIAMOS LOS MEJORES PROGRAMAS! ENVIA A NUESTRO CONCURSO ESE PROGRAMA DEL QUE TE SIENTES ORGULLOSO Y NOSOTROS LO PUBLICAREMOS Y PREMIAREMOS.

BASES

1. Podrán participar todos nuestros lectores, cualquiera sea su edad.
2. Serán aceptados a concurso programas tanto para la primera como para la segunda generación de MSX. Estos programas podrán ser enviados en cinta de cassette, debidamente protegidos en su estuche de plástico, o en disco de 3,5 pulgadas. En este último caso se remitirá al participante un disco virgen a la recepción del programa enviado.
3. Todos los programas deberán llevar la carátula adjunta, o bien fotocopia de la misma.
4. Cada lector puede enviar tantos programas como desee.
5. No se aceptarán programas ya publicados en otros medios o plagiados.
6. Los programas deben seguir las normas usuales de programación estructurada, utilizando líneas REM para marcar todas sus partes, subrutinas donde sean necesarias, etc.
7. Todos los programas deben incluir las correspondientes instrucciones, lista de las variables utilizadas, aplicaciones posibles de programa y todos aquellos comentarios y anotaciones que el autor considere puedan ser de interés para su publicación.

PREMIOS

8. Los programas serán premiados mensualmente, de modo acorde con su calidad, con un premio en metálico de 2.000 a 15.000 ptas.

FALLO Y JURADO

9. El Departamento de Programación de MSX Extra hará la selección de aquellos programas de entre los recibidos según su calidad y su estructuración.
10. Los programas seleccionados aparecerán publicados en la revista MSX Extra, en la que se publicará, junto con el programa, la cantidad con que ha sido premiado.
11. Las decisiones del jurado serán inapelables.
12. Los programas no se devolverán salvo que así lo requiera el autor.

CORTAR O FOTOCOPIAR

TITULO .... **N.º**

TITULO ....

CATEGORIA
PARA .... **K**
INSTRUCCION DE CARGA

AUTOR:
EDAD:
CALLE: .... **N.º** ....
CIUDAD .... **DP** ..... **TEL.:** ....
N.º DE RECEPCION

REMITIR A:
CONCURSO MSX EXTRA
Roca i Batlle, 10-12
bajos
08023 Barcelona

LOS SLOTS (II)

Seguimos con el repaso a los slots que iniciamos en nuestro pasado número. Hoy trataremos algunas nuevas rutinas para el control de los diferentes slots existentes en los MSX, sean de primera o de segunda generación.

En el pasado número indicábamos las rutinas que permitían acceder y modificar el selector de slots de nuestros MSX. De esta forma podemos seleccionar los slots que deseemos de nuestro ordenador.

Pero el primer problema con que nos encontramos al intentar utilizar los slots en diferentes MSX es conocer en cuál de ellos se encuentran los distintos componentes de la memoria: memoria ROM, memoria RAM, etc.

El un MSX, llamémoslo A, la memoria RAM correspondiente a la página 0 puede encontrarse en el slot 2, mientras que en otro MSX, B, puede encontrarse en el slot 1.

Si nosotros lo que queremos es conectar esa zona de RAM, ¿qué slot debemos conectar?

Existe un método seguro para conocer la posición de la RAM, consistente en ir buscándola por todos los posibles SLOTS. Supongamos que queremos localizar la RAM correspondiente a la página 0 de nuestro MSX. Lo primero que haremos es conectar el slot 0 de la página 0. Ahora sólo tenemos que detectar si este slot está ocupado por ROM o por RAM. Para saberlo, escribimos un dato en una posición determinada de memoria, y a continuación lo leemos de la misma. Si dicho dato ha cambiado significa que estamos sobre una zona de memoria ROM (el dato que hemos escrito no ha sido memorizado).

Un consejo: no escribáis datos comunes como 0 ó 255 que puedan encontrarse por casualidad en esa zona de la ROM. Buscad un dato poco usual, como 118 (&H76), que corresponde a la instrucción HALT de ensamblador.

Si hemos localizado RAM, nuestra búsqueda ha terminado; pero si en cambio hemos localizado ROM, hemos de seguir buscando por los slots 1, 2 y 3 hasta localizar la zona de RAM. ¡Atención! Puede que nuestro ordenador no cuente con esa zona de RAM. Estad atentos a todas las posibilidades. Damos un ejemplo de todo este proceso en el listado 1, utilizando para ello nuevas rutinas que comentamos un poco más adelante.

Localizar la ROM es mucho más fácil. En la inmensa mayoría de los MSX, se halla en las páginas 0 y 1 del slot 0. Decimos la inmensa mayoría aunque, en España, son todos. Sin embargo hay que estar siempre alerta. Sobre todo si



pensamos que nuestros programas pueden llegar a otros países.

Si nuestro programa se encuentra en cinta y debe cargarse desde el BASIC, no tenemos ningún problema: la ROM se encuentra en las páginas 0 y 1 que utiliza el BASIC. Sólo tenemos que leer el estado actual del selector de slots.

Si nuestro programa se encuentra en un cartucho ROM, la página 0 en el momento de comenzar la ejecución del programa se halla ocupada por la ROM-BIOS.

El problema aparece cuando utilizamos una unidad de disco y nos encontramos en MSX-DOS o en CP/M. Al cargarse nuestro programa no hallará más que RAM. Para solucionar este problema, y cualquier otro caso en el cual haya dudas, existe una posición de memoria que contiene información acerca de la situación de la ROM-BIOS. Se trata de EXPTBL, que indica el slot en que se encuentra la ROM principal. EXPTBL se encuentra en la dirección &HFCC1, y la forma de interpretar su contenido podéis observarla en la figura 1.

NUEVAS AYUDAS

Si nuestro programa se limita a conectar la configuración de slots más apropiada, y se olvida de los slots a partir de ese momento, no tendremos más problemas.

Pero es muy normal que mientras el programa se encuentra en un slot queramos acceder a datos de otros. Para ayudarnos en este cometido, la ROM-BIOS incorpora toda una serie de rutinas que nos harán mucho más fácil esta tediosa tarea.

* RDSLT (&H000C): Lectura de un byte en un determinado slot.

A <– Slot
HL <– Dirección
A <- Dato leído

Esta rutina permite leer un byte de cualquier posición de memoria de cualquier slot. Para ello basta con indicar en el registro HL la dirección de la que debe leer y en el registro A el slot al que

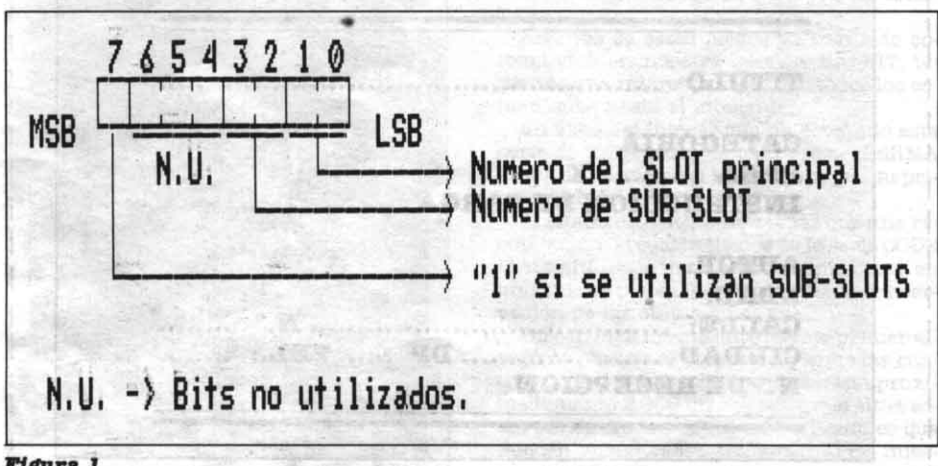


Figura 1

queremos acceder (en el formato de la figura 1).

Si queremos leer datos de la ROM principal debemos colocar a cero los 6 bits de mayor peso, e indicar con los bits 0 y 1 el slot al que queremos acceder.

Modifica los registros AF, BC y DE.

* WRSLT (&H0014): Escritura de un byte en un determinado slot.

A <– Slot
HL <– Dirección
E <– Dato a escribir

Esta rutina permite escribir el byte especificado en el registro E a la dirección indicada por HL, en el slot especificado en el registro A (según el formato de la fig. 1).

Modifica los registros AF, BC y D.

* CALSLT (&H001C): Salto a subrutinas entre slots.

IY <– Slot
IX <– Dirección

Esta rutina realiza un salto a la dirección indicada por IX, del slot que indiquen los 8 bits más altos del registro IY (en el formato de la figura 1). Hay que estar seguros de que en el slot indicado exista la rutina que pretendemos llamar.

Modifica los registros variablemente, dependiendo de la rutina a la que se llame.

* ENASLT (&H0024): Intercambia los slots.

A <– Slot
HL <– Página

Conecta el slot especificado en el registro A (según fig. 1) en la página que indiquen los dos bits de mayor peso del registro HL.

Modifica todos los registros.

* CALLF (&H0030): Salto a subrutinas entre slots.

En este caso la rutina debe ser llamada mediante la instrucción RST &H30, y los parámetros necesarios para su funcionamiento deben encontrarse a continuación de esta instrucción de ensamblador.

Se deben pasar dos parámetros. El primero, un solo byte, indica el slot (según fig. 1) en que se encuentra la rutina que queremos llamar, y el segundo, un word (2 bytes), indica la dirección de memoria a la que queremos que se produzca el salto.

Modifica los registros de forma variable, dependiendo de la rutina a la que se salte.

Es recomendable utilizar RST en lugar de CALL para llamar a esta rutina, ya que de esta forma ahorramos un byte de código, siendo además más rápido el proceso de llamada a dicha rutina.

Incluimos a continuación un ejemplo de uso de esta última rutina. Supongamos que queremos saltar a la dirección 000C de la ROM-BIOS, situada en la página 0 del slot 0. Haríamos:

```
RST   &H30        ; Llamada
 DEFB &B00000000  ; SLOT
DEFW  &H000C      ; DIRECC.
                  ; Resto del programa
```

LISTADO 1

```
PAGE   1
                             ORG   ØCØØØH

               RDSLT:        EQU   ØØØCH
               WRSLT:        EQU   ØØ14H

      CØØØ AF                XOR   A
      CØØ1 3235CØ            LD    (SLOT),A
               PAGØ:
      CØØ4 3A35CØ            LD    A,(SLOT)
      CØØ7 1E76              LD    E,76H
      CØØ9 21FF3F            LD    HL,3FFFH
      CØØC CD14ØØ            CALL  WRSLT
      CØØF 3A35CØ            LD    A,(SLOT)
      CØ12 21FF3F            LD    HL,3FFFH
      CØ15 CDØCØØ            CALL  RDSLT
      CØ18 FE76              CP    76H
      CØ1A 2812              JR    Z,RAMØ
      CØ1C 3A35CØ            LD    A,(SLOT)
      CØ1F 3C                INC   A
      CØ2Ø 3235CØ            LD    (SLOT),A
      CØ23 FEØ4              CP    4
      CØ25 2ØDD              JR    NZ,PAGØ
      CØ27 3E4E              LD    A,"N"
      CØ29 DF                RST   18H
      CØ2A 3E4F              LD    A,"O"
      CØ2C DF                RST   18H
      CØ2D C9                RET

               RAMØ:
      CØ2E 3A35CØ            LD    A,(SLOT)
      CØ31 C63Ø              ADD   A,3ØH
      CØ33 DF                RST   18H
      CØ34 C9                RET

      CØ35 ØØ  SLOT:         DEFB  Ø

                             END
```

UTILIZANDO LOS SLOTS DESDE EL MSX-DOS

El último problema que abordaremos en el capítulo de hoy es el de la utilización de los slots desde el MSX-DOS. En este caso, como ya hemos comentado, el usuario se encuentra con 64 Kb de RAM. Esto provoca que no podamos llamar directamente a las rutinas de la BIOS que controlan los slots.

De este modo llegamos a una paradoja: no podemos llamar a las rutinas que nos permiten cambiar de slots porque están en otro slot al que no podemos acceder sin dichas rutinas.

Pero el MSX está preparado para todo, y en las direcciones inferiores de la RAM utilizada por el MSX-DOS hay una copia de ciertas rutinas que nos permiten realizar trabajos con slots. Estas rutinas son las siguientes:

* RDSLT (&H000C)
* WRSLT (&H0014)
* CALSLT (&H001C)
* ENASLT (&H0024)
* CALLF (&H0030)

La forma de utilizarlas es exactamente la misma que si se tratara de las rutinas de la ROM BIOS anteriormente comentadas.

Y aquí os dejamos por este número. En el próximo incluiremos el último capítulo de esta exposición de la utilidad y funcionamiento de los slots. Hablaremos de los SUB-SLOTS...

por M. Campos

ORDENACION DE FICHEROS 2

En el pasado número hicimos una breve incursión en el mundo de la ordenación de ficheros. Hoy trataremos QUICKSORT, así como el método para ordenar ficheros secuenciales en disco.

QUICKSORT

Como pudimos comprobar en el pasado número, uno de los mayores inconvenientes de las rutinas de ordenación radica en su lentitud. Evidentemente son incomparablemente más rápidas que el proceso habitual de ordenar «a mano»; pero sigue siendo un problema el tener que esperar varios minutos a que se realice la ordenación de nuestro fichero.

Ya indicábamos el mes pasado que existe una rutina que permite realizar ordenaciones de una forma mucho más rápida que las comentadas en aquella ocasión. Se trata de QUICKSORT.

Esta rutina se halla muy extendida con ligeras variantes, ya que cada programador la adecúa a su forma de trabajar. Nosotros incluimos una rutina general, de modo que sea fácil para nuestros lectores adecuarla a sus programas de gestión de ficheros.

QUICKSORT ha sido considerada, desde siempre, una rutina de ordenación complicada. Esto es debido a que el algoritmo que utiliza para realizar la ordenación no es evidente, como ocurría en los métodos por burbuja, sustitución o inserción directa.

Dado que no pretendemos convertir a nadie en un experto programador no explicaremos el algoritmo. Hemos incluido la rutina en el listado 1 para que cualquiera pueda utilizarla sin excesivos quebraderos de cabeza.

Además, para que no haya dudas en cuanto al modo de utilizar esta rutina, incluimos el listado 2, que representa un ejemplo completo de utilización de la misma.

¿COMO ES QUICKSORT?

Quicksort basa su rapidez en una característica muy especial. No ordena el fichero en su totalidad, como hacen las otras rutinas, sino que lo divide en subconjuntos. La ordenación se realiza elemento a elemento entre todos los que forman cada uno de los subconjuntos. Progresivamente se va disminuyendo el tamaño de los subconjuntos hasta que, cuando éstos se convierten en la unidad, el fichero está ordenado totalmente.

Estos subconjuntos se general dividiendo el fichero principal en grupos con un número de elementos potencia de dos. Por ejemplo, en un fichero de 300 elementos (como el caso del ejemplo), se formará un grupo de 256 elementos. Una vez realizada la ordenación elemento a elemento con el resto de elementos del fichero se formarán nuevos grupos de 128 elementos, luego de 64, 32, 16, 8, 4, 2 y finalmente de un solo elemento.

Todo el proceso empleado es complicado, por lo que no profundizaremos más en él; pero aquellos que lo deseen pueden intentar interpretar la rutina (es una tarea algo complicada; pero con un poco de tesón todo es posible).

Pero toda esta complicación sólo tiene una razón de ser, la velocidad. Para resaltar la diferencia de velocidad entre QUICKSORT y el resto de rutinas, hemos realizado una ordenación de un conjunto de 300 elementos utilizando para ello tres rutinas diferentes.

En primer lugar hemos utilizado QUICKSORT, que para ordenar los 300 elementos ha empleado, como media, 14.75 segundos. Hemos de aclarar que este tiempo ha sido calculado utilizando un ordenador compatible IBM AT-286. Podéis observar que no existe ninguna diferencia entre el BASIC de este aparato y el de nuestros MSX. La única diferencia remarcable consiste en que se puede eliminar el parámetro de RND si no se utiliza.

A continuación repetimos el mismo proceso (sobre el mismo ordenador, claro está) con una de las rutinas de SORT comentadas en el pasado número, tardando 114.4 segundos como media en realizar la ordenación de los 300 elementos. La diferencia de tiempos de ejecución es espectacular (del orden de 10 veces más lenta que QUICKSORT).

Los más escépticos criticarán que hayamos utilizado la más lenta de las rutinas comentadas en el pasado número para la medida de la velocidad; pero las razones de esta elección quedarán aclaradas un poco más adelante. Recordad, sin embargo, que los tiempos de ejecución de las tres rutinas comentadas el pasado mes eran bastante similares.

QUICKSORT, ¿UNA PANACEA?

Alguien se preguntará, ¿para qué existen las demás rutinas si ya disponemos de QUICKSORT?

La razón la comentamos ya someramente; no todos los ficheros son iguales. Cada rutina es la mejor para ordenar un tipo de fichero determinado.

Para demostrar esto hemos añadido una segunda medición a las rutinas de los listados 2 y 3. Hemos hecho que la rutina ordene un conjunto de datos ya ordenado. En el caso de QUICKSORT, el tiempo de ejecución se ve reducido a

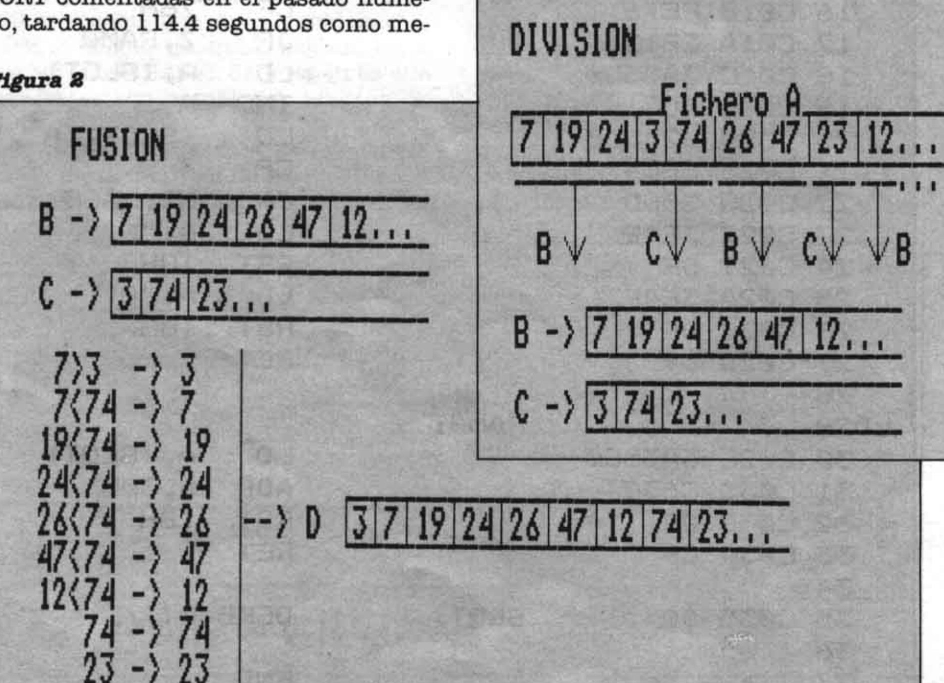


Figura 1

Figura 2

6.45 segundos, mientras que el del SORT normal se mantiene en 97.6 segundos.

Ahora bien, si sabemos que los ficheros a ordenar van a estar casi ordenados podemos modificar la rutina de SORT normal, llegando así al listado 4. En este listado hemos añadido una comprobación: SI el fichero está ordenado ENTONCES hemos terminado de ordenar.

Esta condición es muy fácil de implementar en una rutina de SORT habitual; pero no así en QUICKSORT. Veamos los resultados de la ejecución de los listados 2 y 4.

Si el fichero está totalmente desordenado, la rutina QUICKSORT tarda 14.75 segundos mientras que SORT 2 tarda 126.2 segundos (algo más de 2 minutos).

Peror si el fichero está completamente ordenado QUICKSORT tarda 6.45 segundos, una eternidad comparada con los 0.7 segundos de SORT 2.

Como se demuestra de estas comprobaciones, no hay rutina perfecta, sólo el conocimiento del fichero sobre el que estamos trabajando, y la experiencia, pueden ayudarnos en la difícil elección de la rutina a utilizar en nuestros programas.

LISTADO 1

```
1 '
2 ' *** QUICKSORT ***
3 '
10 L=1:K=0
20 L=L+L:IF L<=MAX THEN 20
30 L=(L-1)\2
40 FOR I=1 TO MAX-L
50    IF A(I)>A(I+L) THEN SWAP A(I),A(I+L):K=I-L
60    IF K<1 THEN 80
70    IF A(K)>A(K+L) THEN SWAP A(K),A(K+L):K=K-L:GOTO 60
80 NEXT I
90 L=L\2:IF L>0 THEN 40
```

LISTADO 2

```
1 '
2 ' *** QUICKSORT ***
3 '
4 ' Tiempo medio:
  14.75 segundos
5 '
6 ' Tiempo ordenado:
   6.45 segundos
7 '
10 DEFINT A-Z
20 MAX=300
30 DIM A(MAX)
40 FOR I=1 TO MAX:A(I)=INT(RND*1000):NEXT I
50 L=1:K=0
60 L=L+L:IF L<=MAX THEN 60
70 L=(L-1)\2
80 FOR I=1 TO MAX-L
90    IF A(I)>A(I+L) THEN SWAP A(I),A(I+L):K=I-L
100    IF K<1 THEN 120
110    IF A(K)>A(K+L) THEN SWAP A(K),A(K+L):K=K-L:GOTO 100
120 NEXT I
130 L=L\2:IF L>0 THEN 80
```

LISTADO 3

```
1 '
2 ' *** SORT ***
3 '
4 ' Tiempo medio:
 114.4 segundos
5 '
6 ' Tiempo ordenado:
  97.6 segundos
7 '
10 DEFINT A-Z
20 MAX=300
30 DIM A(MAX)
40 FOR I=1 TO MAX:A(I)=INT(RND*1000):NEXT I
50 FOR I=1 TO MAX-1
60    FOR J=I TO MAX
70       IF A(I)>A(J) THEN SWAP A(I),A(J)
80    NEXT J
90 NEXT I
```

LISTADO 4

```
1 '
2 ' *** SORT 2 ***
3 '
4 ' Tiempo medio:
 126.2 segundos
5 '
6 ' Tiempo ordenado:
   0.7 segundos
7 '
10 DEFINT A-Z
20 MAX=300
30 DIM A(MAX)
40 FOR I=1 TO MAX:A(I)=INT(RND*1000):NEXT I
50 FOR I=1 TO MAX-1
60    FOR J=I TO MAX
70       IF A(I)>A(J) THEN SWAP A(I),A(J):N=1
80    NEXT J
90 NEXT I
```

ORDENACION DE FICHEROS SECUENCIALES

Lo prometido es deuda y, como habíamos anunciado en nuestro pasado número, explicaremos el modo de realizar ordenaciones de ficheros secuenciales.

Tal vez no quedó suficientemente claro en su momento, y alguien abrigó falsas esperanzas; pero es imposible ordenar un fichero secuencial en cinta.

Esta espantosa limitación viene provocada por la imposibilidad de mantener dos ficheros abiertos al mismo tiempo en el cassette. No podemos por tanto leer datos de un fichero para grabarlos en otro una vez ordenados.

Lo que sí podemos hacer es cargar el fichero secuencial de la cinta en memoria. Una vez hecho esto podemos aplicar cualquiera de las rutinas de ordenación que ya conocemos, sin ningún problema, y grabar el resultado en otro fichero en cinta.

Lamentablemente esto no lo podremos realizar con los ficheros más grandes, que excedan la capacidad de memoria de la máquina.

Sólo los usuarios de unidad de disco podrán realizar este tipo de ordenación en sus ficheros secuenciales.

PARA ORDENAR...

Al contrario de lo que hemos hecho con la rutina QUICKSORT, aquí expondremos el método a seguir para realizar la ordenación; pero no daremos listado. Esperamos que esto sirva de ejercicio para nuestros sufridos lectores.

El proceso a seguir es muy simple, hay que dividir el fichero en dos (algo más ordenados que el primero), y luego volver a fusionarlos en un fichero. Hay que repetir el proceso tantas veces como sea necesario.

DIVISION

Llamaremos, para clarificar el proceso, fichero A al fichero original, y ficheros B y C a los ficheros obtenidos de la división del fichero A. El proceso de división del fichero en dos es el siguiente. Hay que enviar al fichero B el

primer elemento del fichero A. Se compara con este elemento el segundo del fichero A. Si resulta estar correctamente ordenado se envía al fichero B y se repite el proceso con el siguiente elemento del fichero A.

Si el elemento no está correctamente ordenado, se envía al fichero C. Se recoge el siguiente elemento de A, y en caso de estar correctamente ordenado se transfiere a C (cuidado). Se repite el proceso hasta encontrar un elemento desordenado, que se vuelve a enviar a B.

Sólo hay que repetir este proceso hasta que se hayan repartido todos los elementos del fichero A.

Para entender mejor este proceso, os remitimos a la figura 1.

FUSION

La fusión no es más complicada que la división. Llamaremos D al nuevo fichero, resultado de la fusión de B y C.

Tomamos el primer elemento de B y el primero de C. Enviamos el menos de ellos a D (si ordenamos de menor a mayor), y leemos otro elemento del fichero al que perteneciera el elemento grabado en D. Se repite esta comparación hasta que se haya fusionado la totalidad de los ficheros.

Para entender mejor este proceso os remitimos a la figura 2.

TERMINANDO

El último problema con que nos encontramos radica en saber cuándo se ha terminado la ordenación. La solución es muy sencilla. Cuando el fichero A está ordenado, el fichero B es igual al fichero A y el fichero C está vacío. ¿Os animáis a programarlo vosotros?

Y LA TRISTE DESPEDIDA

Terminamos aquí el artículo de hoy, y con él esta sección, «Tratamiento de ficheros». Esperamos que haya servido para que conozcáis un poco mejor el fascinante mundo de los ficheros.

Esperamos que no haya sido inconveniente el dejar muchos apartados como ejercicios. Hemos creído que esta solución era interesante en un tema como los ficheros, de por sí algo aburrido. Hemos preferido que os peleéis un poco con vuestro ordenador a que tengáis que soportar parrafadas interminables de texto.

¡Y eso sí!, aunque acabe esta sección, no nos olvidaremos de los usuarios de unidades de disco, ni de aquellos que busquen algo más que juegos para su MSX.

Por Willy Miragall

POKES DE VIDAS INFINITAS

Muchos de vosotros os habréis preguntado cómo se pueden encontrar esos «mágicos» pokes de vidas infinitas, energía perpetua, ver el final del juego y un largo etcétera. En este artículo vamos a intentar desvelar los misterios de cómo encontrarlos.

En un principio, encontrar un poke de vidas infinitas es casi una labor detectivesca, que se hace más o menos fácil según la forma que el autor del juego lo haya protegido.

Para poder analizar un juego se necesita primero un «equipo» compuesto por los siguiente programas:

— Un desensamblador: Indispensable para poder ir descifrando las rutinas que nos llevarán a encontrar el tan ansiado poke. El desensamblador debería ocupar poca memoria y poder ser relocalizable, y a la vez ser completamente manejable. No es necesario que nos proporcione un listado «fuente» del programa, sino que se limite a ir traduciendo el programa a medida que se lo pidamos.

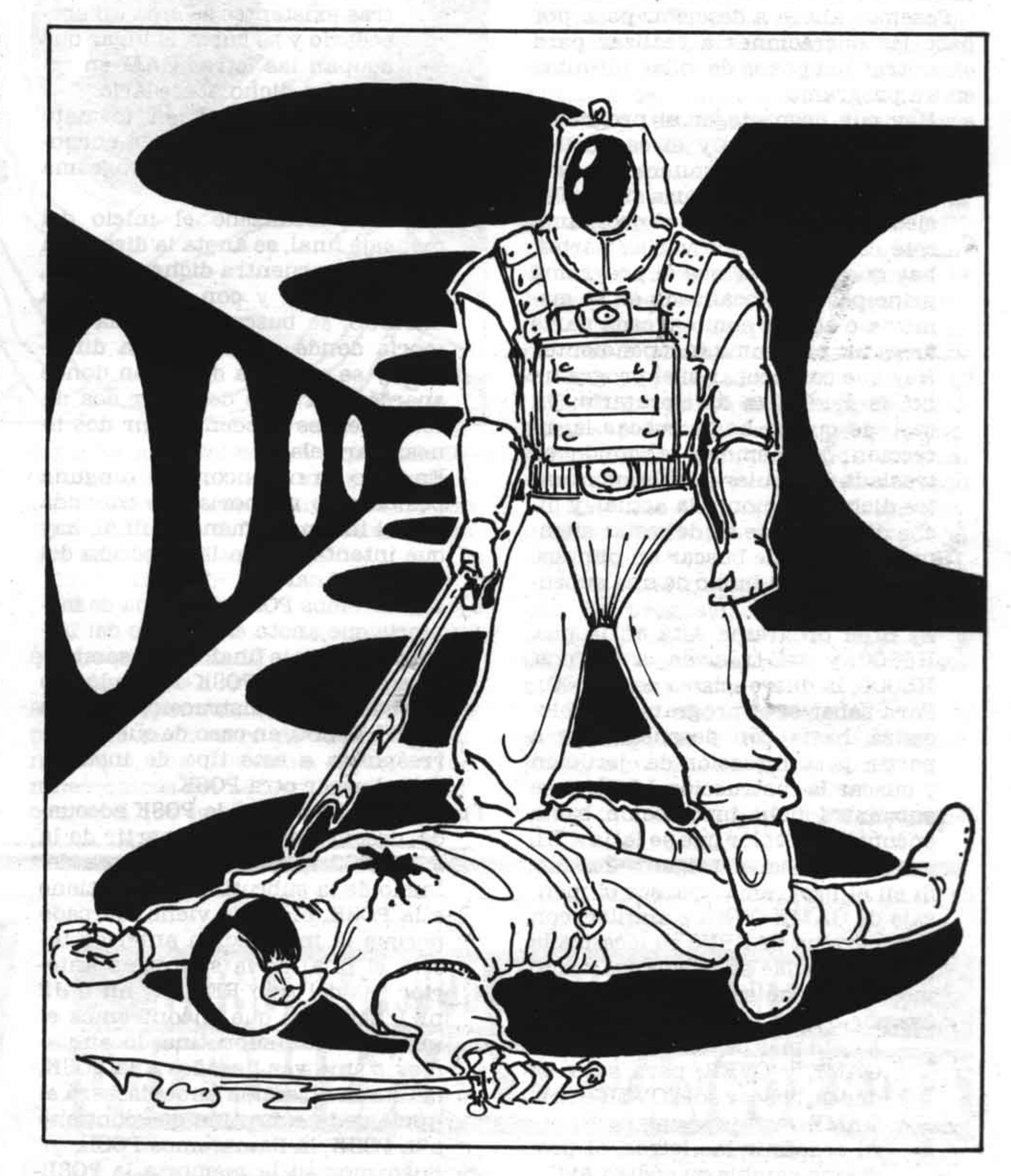
— Un programa que nos dé las direcciones de inicio, final y ejecución de un programa (CASNOM) para poderlo analizar.

— Un programa de búsqueda (CARBUS): Este programa tiene la función de buscar una cadena de caracteres en la memoria, unos bytes determinados o una dirección determinada. Es fácil de hacer un programa de estas características en Basic, pero sería demasiado lento, por lo que es más útil hacerlo en lenguaje máquina. Es también muy interesante que dicho programa pueda buscar bytes relativos, es decir, si el primer byte que recoge vale X, el segundo debe valer X-6 y el tercero X+6, independientemente del valor X.

— Un programa que nos presente la memoria en formato 'sprite' (BUSCA): Sirve principalmente para localizar mensajes que no estén escritos con caracteres, sino que estén almacenados en formato gráfico. Tiene además una doble utilidad, ya que nos permite localizar los 'sprites' de un programa, borrar a los enemigos (poniendo ceros en las posiciones de memoria que los definen) y si el programa se limita a comprobar si hay colisiones de sprites, ya se habrán anulado a los enemigos. Este método puede ser utilizado principalmente en programas como Manic Miner, Blagger, Jet Set Willy, etc...

— Un programa que visione toda la memoria en formato de texto (LETBUS): Permitirá localizar todos los mensajes de un programa (si los caracteres no están 'camuflados') y obrar en consecuencia.

Es también indispensable tener un cierto conocimiento de ensamblador para poder localizar el inicio y final de cada subrutina.



Para aclarar un poco toda la explicación de cómo encontrar un poke, vamos primero a explicar el motivo por el que se hace así:

Pongámonos por un momento en lugar del programador y pensemos cómo haríamos nosotros que nos quitasen una vida.

¿Le ha tocado la bomba? (esta comprobación puede ser muy complicada)

Si no es así, vuelve.

Llama a la subrutina de quitar una vida (SUBVID).

Vuelve.

SUBVID: Decrementa el número de vidas (una posición de memoria).

Compara el número de vidas con 0.

Si no es 0, vuelve.

Salta a subrutina de imprimir mensaje final (MENFIN).

Vuelve.

MENFIN: Carga en un valor el inicio del mensaje final (LD HL,nn), llama subrutina que imprime el mensaje que se encuentre en HL, salta al inicio del programa.

Lo que es fundamental es lograr acceder a una subrutina que se ejecute cuando se nos reste una vida o se acabe

el juego. Entonces todo se limita a saltar de una subrutina a otra hasta encontrar la que decrementa el número de vidas o hace la comparación con 0.

El ejemplo anteriormente puesto es una simplificación de un programa real, y lo único que se hace, es una vez localizada MENFIN, buscar quién llama a MENFIN (SUBVID) y modificarla de manera que nunca llame a MENFIN.

Pasemos ahora a describir paso por paso las operaciones a realizar para encontrar los pokes de vidas infinitas en un programa:

a) Hay que desproteger el programa (si está protegido y si es posible desprotegerlo) de manera que esté localizado en la memoria sin que se ejecute. En caso de que el programa esté compuesto por varias partes, hay que procurar que el programa principal esté localizado en la memoria o actuar como si cada parte fuera un programa independiente.

b) Hay que comprobar si el programa no se relocaliza al ejecutarlo, en caso de que lo haga, buscar la dirección de memoria a donde se traslada y calcular la diferencia entre dicha posición y la actual, y dicha diferencia se ha de restar siempre a la hora de buscar un puntero o la llamada al inicio de una subrutina.

   Ej. Si el programa está en la pos. H8800 y se traslada a la pos. H8000, la diferencia es de H0800.

   Para saber si el programa se relocaliza, basta con desensamblar a partir de su dirección de ejecución y buscar la instrucción LDIR, si se encuentra dicha instrucción, basta encontrar el valor que se le da a HL y a DE para saber origen y destino.

c) Si en el programa aparece el mensaje de GAME OVER o similar, con el programa CARBUS se localiza la posición donde se encuentra dicho mensaje. Si no se encuentra puede ser por 3 motivos:

   1— Existe más de un espacio entre GAME y OVER, para salir de dudas, buscar sólo OVER o sólo GAME.

   2— Al redefinir las letras, el programa cambia su código ASCII (las camufla), de manera que hay que buscar el mensaje según caracteres relativos, para ello se averigua el código ASCII de las tres primeras letras del mensaje (en caso de GAM, son 71,65 y 77). A continuación se busca un byte cuyo siguiente sea su igual menos 6, y el siguiente del siguiente sea su igual más seis, esto traducido a Basic sería:

```
FOR I=Dir. inicio TO Dir. final:
A=PEEK(I):   B=PEEK(I+1):
C=PEEK (I+2): IF A-B=6 AND
C-A=6 THEN PRINT I: NEXT
ELSE NEXT
```

   Si no se encuentra el mensaje puede ser porque al redefinir los caracteres, no redefine todo el abecedario sino que se limita a redefinir los caracteres que necesita. Para saber qué caracteres son estos, con el programa BUSCA, se localizan las letras redefinidas, y con las letras existentes se crea un abecedario y se busca el lugar que ocupan las letras GAM en relación a dicho abecedario.

   3— El mensaje está en formato gráfico, siendo fácil de encontrar entonces con el programa BUSCA.

d) Una vez localizado el inicio del mensaje final, se anota la dirección donde se encuentra dicho mensaje, seguidamente, y con el programa CARBUS, se busca en toda la memoria donde aparece dicha dirección y se anota la dirección donde aparece. En caso de haber dos direcciones, se pueden seguir dos líneas paralelas.

   En caso de no encontrar ninguna posición de memoria que coincida con el inicio del mensaje final, hay que intentarlo con la dirección del mensaje final −1, −2, etc...

e) Llamaremos POSK a la zona de memoria que anota el puntero del inicio del mensaje final. Desensamblamos a partir de POSK-1, y lo lógico es obtener una instrucción del tipo LD HL,43250, en caso de que no corresponda a este tipo de instrucción, buscar otra POSK.

f) Una vez localizada la POSK adecuada, desensamblamos a partir de la POSK-200 o similar en busca del inicio de la subrutina que contiene a la POSK, el inicio viene marcado porque la instrucción anterior indica el final de la subrutina anterior (es del tipo RET, JR nn o JP nn). Cada vez que encontramos el inicio de una subrutina, lo anotamos, y una vez llegados a la POSK, la última dirección anotada, será el inicio de la subrutina que contiene a la POSK, la llamaremos POSL.

g) Buscamos en la memoria la POSL, llamándola POSK, y desciframos a partir de POSK-1, en caso de encontrar una llamada o salto relativo (del tipo CALL Z,nn, JP NC,nn...), ponemos ceros en el lugar donde se encuentre dicha instrucción y ejecutamos el juego, si ha habido suerte, al llegar a 0 vidas no saldrá el mensaje de GAME OVER y podremos seguir jugando, en caso contrario, pasamos al punto f.

   En caso de no encontrar la POSL, puede ser porque no es la POSL adecuada o porque la llamada que se le hace se hace mediante un salto relativo (JR nn). Para encontrar la dirección donde hace dicho salto, habrá que usar el desensamblador, y desensamblar desde POSL-128 hasta POSL+128, buscando una instrucción JR que llame a la POSL. Una vez encontrada dicha instrucción, procedemos como indica el principio de este punto.

h) Una variante puede ser la de, al encontrar una subrutina, buscar si en dicha subrutina se incrementa, compara u opera con alguna dirección (o con registros del tipo A o [HL]). En este caso, basta buscar en toda la memoria (partiendo de la base que la dirección que tenemos contiene el número de vidas) la dirección, encontrando así donde se adjudica el número de vidas iniciales:

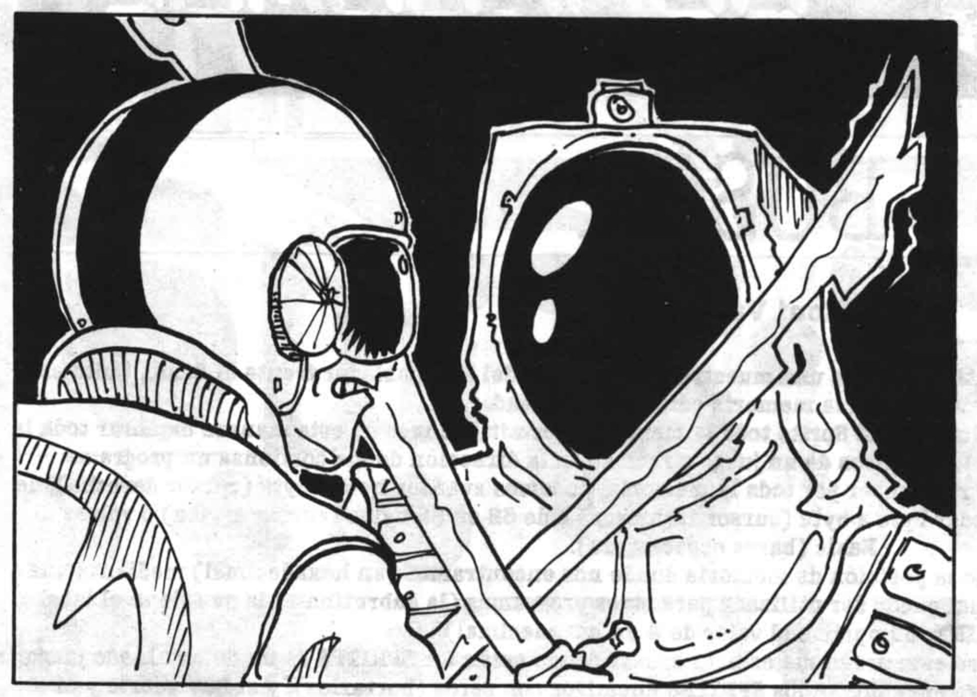
   Ej. POSL LD A,(34769) 'Posible dirección que contiene las vidas. CP 0
   POSK-1:JR Z,43401 'Dirección que saca el mensaje GAME OVER. JP 42010

   Una forma de comprobar si una dirección contiene el número de vidas es la de ejecutar el juego, y a un número determinado de vidas, pulsar el botón de RESET si vuestro MSX dispone de él y comparar el número de éstas con el número que contiene la posición de memoria que tenemos.

Pasemos ahora a tratar un ejemplo completo. Se trata del juego MAXIMA, que para los que no lo sepan, es un típico juego de marcianos que se adapta perfectamente a lo explicado hasta ahora. Vamos a explicar todo el proceso seguido para encontrar un POKE de vidas infinitas.

Este juego tiene una protección bastante buena, ya que carga un programa en Basic con BLOAD para ejecutarlo seguidamente, cargando entonces la segunda parte que es el juego propiamente dicho. Es esta la parte sobre la que centraremos nuestras pesquisas:

Para empezar, encontramos fácilmente el mensaje GAME OVER, que se encuentra en la posición de memoria (39592). Buscamos entonces la posición de memoria que contiene esta dirección, y obtenemos la 40418. Pasamos entonces a buscar el inicio de la subrutina que contiene al 40418, y resulta la dirección (40379). Buscamos a continuación la zona de memoria que contiene esta dirección, obteniendo la 40321. Descifrando la instrucción que hace la llamada, comprobamos que es un salto sólo si un valor es igual a 0. Muy animados, introducimos 3 pokes (POKE 40320,0: POKE 40321,0: POKE 40322,0) y ejecutamos el juego, dejamos que nos maten todas las vidas y esperamos con el corazón en vilo; pero, ¡oh, decepción!, no sale el mensaje de GAME OVER, pero sí que se acaba la partida. Continuamos nuestras pesqui-



sas desde el punto que las dejamos, es decir, en la posición 40321, buscamos el inicio de la subrutina que contiene a esta instrucción, y la obtenemos: es la (40318). Pasamos a buscar la zona de memoria que contiene dicha dirección, y damos con la 37628. Descifrando esta instrucción, comprobamos que se trata de otro salto dependiente del valor de un registro. Un poco más animados, introducimos otros 3 pokes (POKE 37627,0: POKE 37628,0: POKE 37629,0) y ejecutamos el juego y... ¡oh, maravilla de las maravillas! Después de matar todos los tanques seguimos jugando como tal cosa. Ya han sido localizados los pokes de vidas infinitas; pero aún se puede hilar más fino. He aquí un listado de la zona de memoria que hemos poqueado:

```
LD A,(IY+17)
OR A
JP M,40318 'Esta instrucción es la que ha sido anulada.
```

Pocas dudas nos quedan de que la posición de memoria (IY+17) es la que contiene el número de vidas, de manera que, con una tabla de todos los mnemónicos del Z80 en la mano, buscamos el que cargue en (IY+d) un valor, y obtenemos que tiene 3 bytes que lo definen. Estos son:

253 (fijo), n (indica si se carga con A, B, C...) y 17 (el valor que se le suma a IY).

Pasamos a hacer uso del CARBUS, buscamos un 253, comodín (cualquier valor, se introduce con un # o con un 35 en decimal) y un 17, y obtenemos 6 posiciones de memoria (37494, 37623, 38015, 38026, 40327 y 40379), y nos dedicamos a desensamblarlas una por una, encontrando que dos de ellas se limitan a decrementar el contenido de (IY+17), estas son las direcciones 38015 y la 40379. Cogemos la primera y le introducimos un solo poke (POKE 38015,17), con el que cambiamos la instrucción, que pasa de ser DEC (IY+17) a LD DE,4405, con lo que queda anulado su efecto. Ejecutamos el juego y efectivamente, el número de vidas no decrece. Después de dar un par de vueltas al marcador, pasamos al siguiente y último apartado.

El último punto es el de cómo introducir el poke en el programa. Para ello, basándonos en el programa original (reduciéndolo un poco), hemos creado un sustituto del primer trozo del programa, de manera que sólo cargue el segundo.

Y ya para finalizar, sólo falta decir que los programas citados anteriormente (BUSCA, CARBUS, LETBUS y CASNOM) aparecen publicados en este mismo número.

Si lográis encontrar algún poke de vidas infinitas o similar para algún juego, no dudéis en enviarnoslo.

Roni Van-Ginkel



BUSCA

Roni Van-Ginkel

El programa que os ofrecemos a continuación es una muestra de la ventaja del Ensamblador frente al Basic, tanto en ahorro de la memoria como en velocidad.

El programa en sí se limita a pasar a formato de Sprite toda la memoria, permitiéndonos de esta manera explorar toda la memoria del ordenador en busca de los gráficos de un juego o incluso de la dirección donde comienza un programa.

Este programa nos permite avanzar y retroceder por toda la memoria, podemos avanzar byte a byte (cursor derecha), de 32 en 32 bytes (cursor abajo), retroceder byte a byte (cursor izquierda) o de 32 en 32 bytes (cursor arriba) o volver al Basic (barra espaciadora).

También nos indica en cada momento la posición de memoria donde nos encontramos (en hexadecimal) mediante una corta subrutina en ensamblador, la cual puede ser utilizada para otros programas (la subrutina es la que tiene el label PRINT, e imprime el valor de A en hexadecimal).

Personalmente, para lo que más utilizo este programa es para buscar dónde están los SPRITES de un determinado juego, rellenar la zona de memoria donde se encuentran los SPRITES enemigos con ceros (borrarlos), y si hay suerte y el programa sólo detecta las colisiones se habrán anulado todos los enemigos, que quieras que no, siempre resulta una buena ayuda.

Antes de ejecutar el programa es conveniente hacer un SCREEN 1,2 para poner un tamaño de los sprites adecuado.

Como el programa está en la zona de memoria que utiliza el Basic para almacenar sus programas, no es aconsejable cargarlo cuando existe un programa en memoria, por el mismo motivo, el listado en Basic para cargar el programa en memoria que adjuntamos tiene que cargar primero el programa en una zona libre de memoria, hacer un NEW (os aconsejamos grabar el programa antes de hacer un RUN) y trasladar el programa hacia su zona de memoria correcta. Una vez hecho esto, podéis grabar el programa con un BSAVE "BUSCA", 32800, 33075

y para cargarlo:

BLOAD "BUSCA": DEFUSR=32800: KEY 2, "D=USR(0)" + CHR$(13)

con lo que al pulsar la tecla F2, se ejecutará el programa.

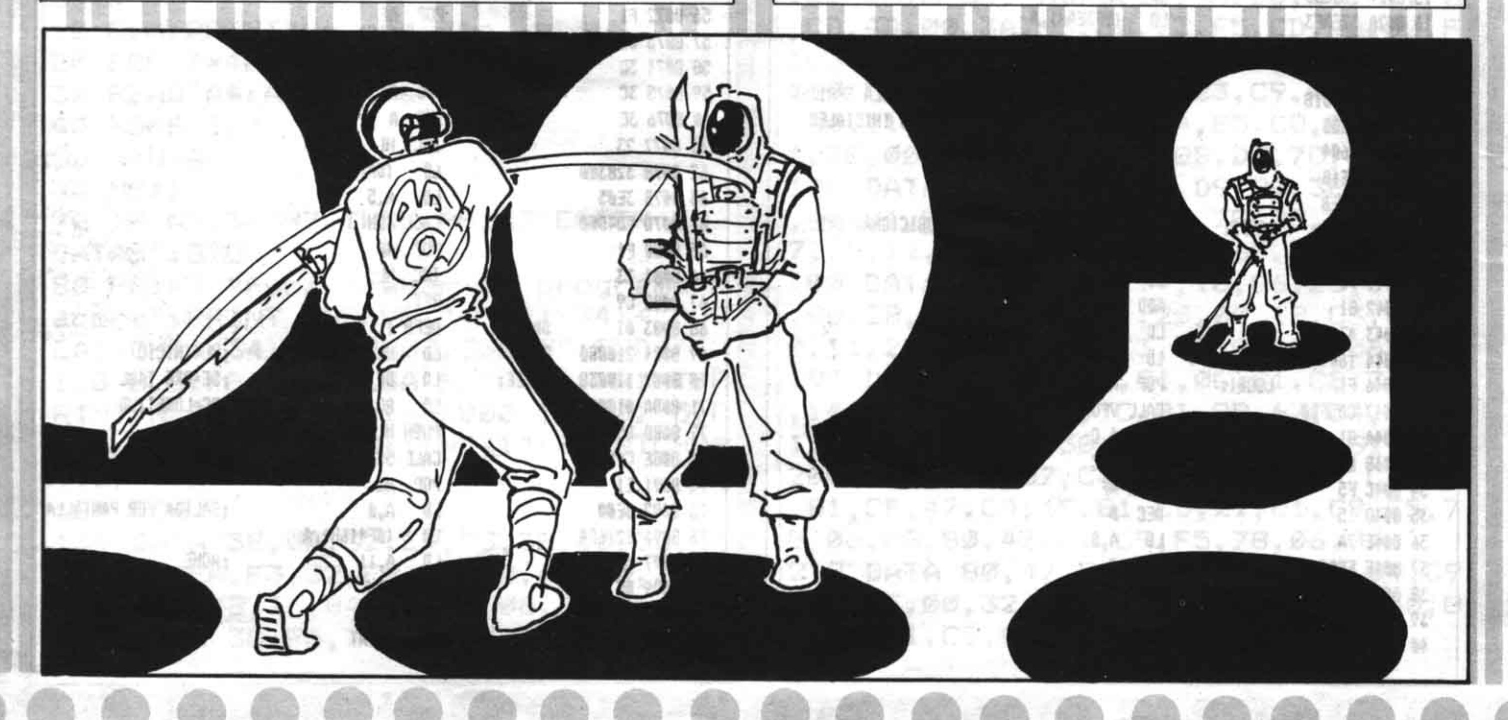
```
1 '
2 ' *** Cargador BUSCA ***
3 '
10 CLS:PRINT"Un momento ..."
20 FOR I=40000! TO 40275!
30 READ A$:A=VAL("&H"+A$)
40 POKE I,A
50 N=N+A
60 NEXT
70 IF N<>30797 THEN PRINT"ERROR EN
DATAS":STOP
80 PRINT "Para grabar el programa h
acer:":PRINT "BSAVE "+CHR$(34)+"BUS
CA"+CHR$(34)+",32800,33075":END
110 CLS:PRINT"PULSA RETURN":PRINT:P
RINT"N=32800:FOR I=40000 TO 40275:P
OKE N,PEEK(I):N=N+1:NEXT":PRINTCHR$
(11)
120 NEW
130 DATA 3E,0A,32,E9,F3,3E,00,32,83
,80,32,EA,F3,32,EB,F3,CD,6F,00,21,0
0,1B,1E,08,16,04,0E,18,06,EB,3E
140 DATA 3C,F5,78,81,47,16,04,F1,CD
,5F,80,81,81,F5,15,7A,FE,00,C2,46,8
0,F1,1D,7B,FE,00,C2,3E,80,C3,84
150 DATA 80,F5,78,CD,4D,00,23,F1,F5
,CD,4D,00,3A,83,80,23,F5,CD,4D,00,F
1,3C,3C,3C,3C,23,32,83,80,3E,05
160 DATA CD,4D,00,F1,23,C9,01,21,00
,80,11,00,38,01,00,04,E5,CD,5C,00,E
1,3E,00,32,16,F4,3E,0B,DF,7C,CD
170 DATA D9,80,7D,CD,D9,80,3E,08,CD
,41,01,FE,FF,28,F7,CB,7F,28,12,CB,6
7,28,12,CB,77,28,19,CB,6F,28,0E
180 DATA CB,47,28,19,18,C6,23,C3,87
,80,2B,C3,87,80,11,20,00,19,C3,87,8
0,11,20,00,ED,52,C3,87,80,C9,06
190 DATA 30,CB,7F,C4,0C,81,CB,77,C4
,14,81,CB,6F,C4,1C,81,CB,67,C4,1F,8
1,CD,21,81,06,30,CB,5F,C4,0C,81
200 DATA CB,57,C4,14,81,CB,4F,C4,1C
,81,CB,47,C4,1F,81,CD,21,81,C9,F5,7
8,06,08,80,47,F1,C9,F5,78,06,04
210 DATA 80,47,F1,C9,04,04,C9,04,C9
,F5,3E,00,32,16,F4,78,FE,3A,D4,30,8
1,DF,F1,C9,06,07,80,C9
```




```
; *** BUSCA ***
;
; Programa que presenta toda
; la memoria central en formato
; SPRITE para localizar los
; gráficos de un programa .
;
; Por Roni Van-Ginkel
;
                ORG  32800       ;DIRECCION INICIO
                LOAD 32800
8020 3E0A        LD   A,10        ;COLOR 10,1,1
8022 32E9F3      LD   (0F3E9H),A
8025 3E00        LD   A,0
8027 328380      LD   (DAT),A
802A 32EAF3      LD   (0F3EAH),A
802D 32EBF3      LD   (0F3EBH),A
8030 CD6F00      CALL 6FH         ;SCREEN 1
8033 21001B      LD   HL,6912     ;ORG TABLA SPRITES
8036 1E08        LD   E,8         ;DATOS INICIALES
8038 1604        LD   D,4
803A 0E18        LD   C,24
803C 06EB        LD   B,235
803E 3E3C LOOP2: LD   A,60        ;POSICIONAR SPR.
8040 F5          PUSH AF
8041 78          LD   A,B
8042 81          ADD  A,C
8043 47          LD   B,A
8044 1604        LD   D,4
8046 F1   LOOP1: POP  AF
8047 CD5F80      CALL VPOKE
804A 81          ADD  A,C
804B 81          ADD  A,C
804C F5          PUSH AF
804D 15          DEC  D
804E 7A          LD   A,D
804F FE00        CP   0
8051 C24680      JP   NZ,LOOP1
8054 F1          POP  AF
8055 1D          DEC  E
8056 7B          LD   A,E
8057 FE00        CP   0
8059 C23E80      JP   NZ,LOOP2
805C C38480      JP   PROG
805F F5   VPOKE: PUSH AF          ;ESCRIBIR DATOS
8060 78          LD   A,B         ;EN LA VRAM
8061 CD4D00      CALL 4DH
8064 23          INC  HL
8065 F1          POP  AF
8066 F5          PUSH AF
8067 CD4D00      CALL 4DH
806A 3A8380      LD   A,(DAT)
806D 23          INC  HL
806E F5          PUSH AF
806F CD4D00      CALL 4DH
8072 F1          POP  AF
8073 3C          INC  A
8074 3C          INC  A
8075 3C          INC  A
8076 3C          INC  A
8077 23          INC  HL
8078 328380      LD   (DAT),A
807B 3E05        LD   A,5
807D CD4D00      CALL 4DH
8080 F1          POP  AF
8081 23          INC  HL
8082 C9          RET
8083 01   DAT:   DEFB 1
8084 210080 PROG:  LD HL,32768    ;HL=INICIO
8087 110038 BUCLE: LD DE,14336    ;DE=ORG TABL
808A 010004      LD   BC,1024     ;BC=LONGITUD
808D E5          PUSH HL
808E CD5C00      CALL 5CH         ;DEFINIR SPRITES
8091 E1          POP  HL
8092 3E00        LD   A,0         ;SALIDA POR PANTALLA
8094 3216F4      LD   (0F416H),A
8097 3E0B        LD   A,11        ;HOME
8099 DF          RST  18H
809A 7C          LD   A,H
809B CDD980      CALL PRINT
```




```
809E 7D                 LD   A,L
809F CDD980             CALL PRINT
80A2 3E08      TECLA:   LD   A,8            ;TECLADO
80A4 CD4101             CALL 141H
80A7 FEFF               CP   255            ;PULSACION ?
80A9 28F7               JR   Z,TECLA        ;SI NO ESPERA
80AB CB7F               BIT  7,A
80AD 2812               JR   Z,RIGHT        ;CURSOR DERECHA
80AF CB67               BIT  4,A
80B1 2812               JR   Z,LEFT         ;CURSOR IZQUIERDA
80B3 CB77               BIT  6,A
80B5 281A               JR   Z,DOWN         ;CURSOR ABAJO
80B7 CB6F               BIT  5,A
80B9 280E               JR   Z,UP           ;CURSOR ARRIBA
80BB CB47               BIT  0,A
80BD 2819               JR   Z,EXIT         ;BARRA = SALIR
80BF 18C6               JR   BUCLE
80C1 23        RIGHT:   INC  HL             ;AVANZAR LENTO
80C2 C38780             JP   BUCLE
80C5 2B        LEFT:    DEC  HL             ;RETROCEDER LENTO
80C6 C38780             JP   BUCLE
80C9 111F00    UP:      LD   DE,31          ;RETROCEDER RAPIDO
80CC ED52               SBC  HL,DE
80CE C38780             JP   BUCLE
80D1 112000    DOWN:    LD   DE,32          ;AVANZAR RAPIDO
80D4 19                 ADD  HL,DE
80D5 C38780             JP   BUCLE
80D8 C9        EXIT:    RET                 ;VOLVER AL BASIC
80D9 0630      PRINT:   LD   B,48           ;IMPRIME EL
80DB CB7F               BIT  7,A            ; VALOR DE A EN
80DD C40C81             CALL NZ,P8          ;HEXADECIMAL
80E0 CB77               BIT  6,A
80E2 C41481             CALL NZ,P4
80E5 CB6F               BIT  5,A
80E7 C41C81             CALL NZ,P2
80EA CB67               BIT  4,A
80EC C41F81             CALL NZ,P1
80EF CD2181             CALL PRINT2
80F2 0630               LD   B,48
80F4 CB5F               BIT  3,A
80F6 C40C81             CALL NZ,P8
80F9 CB57               BIT  2,A
80FB C41481             CALL NZ,P4
80FE CB4F               BIT  1,A
8100 C41C81             CALL NZ,P2
8103 CB47               BIT  0,A
8105 C41F81             CALL NZ,P1
8108 CD2181             CALL PRINT2
810B C9                 RET
810C F5        P8:      PUSH AF
810D 78                 LD   A,B
810E 0608               LD   B,8
8110 80                 ADD  A,B
8111 47                 LD   B,A
8112 F1                 POP  AF
8113 C9                 RET
8114 F5        P4:      PUSH AF
8115 78                 LD   A,B
8116 0604               LD   B,4
8118 80                 ADD  A,B
8119 47                 LD   B,A
811A F1                 POP  AF
811B C9                 RET
811C 04        P2:      INC  B
811D 04                 INC  B
811E C9                 RET
811F 04        P1:      INC  B
8120 C9                 RET
8121 F5        PRINT2:  PUSH AF
8122 3E00               LD   A,0
8124 3216F4             LD   (0F416H),A
8127 78                 LD   A,B
8128 FE3A               CP   58
812A D43081             CALL NC,SUMA
812D DF                 RST  18H
812E F1                 POP  AF
812F C9                 RET
8130 0607      SUMA:    LD   B,7
8132 80                 ADD  A,B
8133 C9                 RET
                        END
```



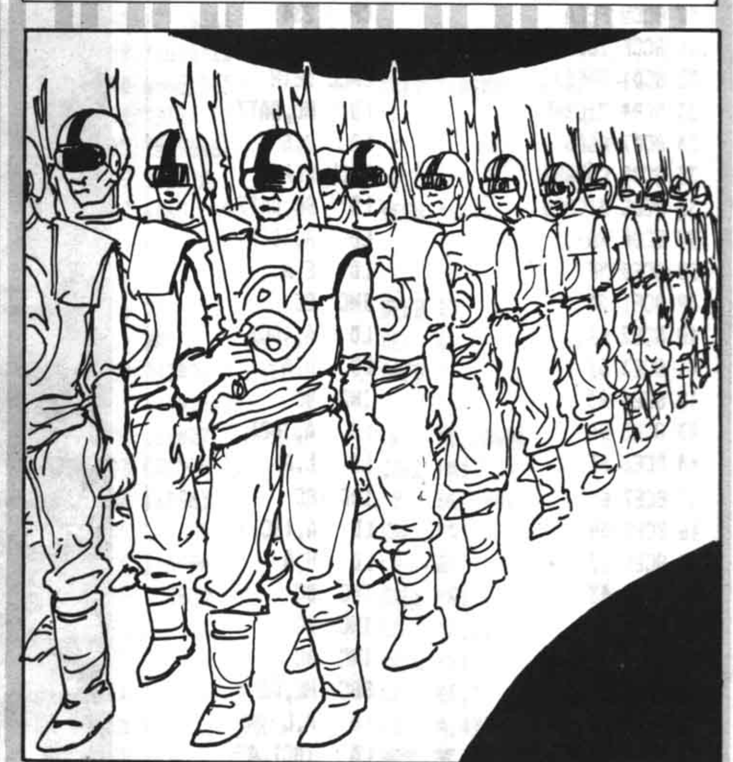
CASNOM

Por Roni Van-Ginkel

El programa que ahora os ofrecemos sirve para conocer las direcciones de inicio, final, ejecución y la longitud de un programa grabado en cinta (en caso de ser un programa en Basic, se limita a poner el nombre y el formato con que ha sido grabado).

Para lanzar el programa, basta poner DEFUSR=36000: D=USR(0) y poner la cinta en el principio, pudiéndose interrumpir la carga en cualquier momento pulsando CTRL+STOP.

```
1 '
2 ' Cargador para el programa CASNO
M
3 '
1Ø CLS:PRINT"Un momento ..."
2Ø FOR I=36ØØØ! TO 36347!
3Ø READ A$
4Ø A=VAL("&H"+A$)
5Ø POKE I,A
6Ø N=N+A
7Ø NEXT
8Ø IF N<>34758! THEN PRINT"ERROR EN
 DATAS":STOP
9Ø PRINT "Para grabar hacer:":PRINT
 "BSAVE "+CHR$(34)+"CASNOM"+CHR$(34
)+",36ØØØ,3635Ø":END
1ØØ DATA CD,E1,ØØ,CD,E4,ØØ,32,C7,8D
,CD,E4,ØØ,FE,CF,3Ø,F9,21,53,8D,77,2
3,Ø6,Ø5,CD,2A,8D,CD,E7,ØØ,21,4C
11Ø DATA 8D,CD,2Ø,8D,CD,15,8D,3A,C7
,8D,FE,D3,28,7Ø,FE,EA,28,67,CD,E1,Ø
Ø,21,C8,8D,Ø6,Ø6,CD,2A,8D,Ø1,C8
12Ø DATA 8D,ØA,5F,Ø3,ØA,57,Ø3,ØA,6F
,Ø3,ØA,67,Ø3,Ø3,Ø3,ED,52,7D,Ø2,Ø3,7
C,Ø2,11,C8,8D,21,5A,8D,Ø6,Ø4,E5
13Ø DATA E1,CD,2Ø,8D,E5,1A,6F,13,1A
,67,13,D5,CD,DØ,8D,CD,15,8D,D1,1Ø,E
B,E1,18,34,3E,ØD,CD,18,ØØ,3E,ØA
14Ø DATA CD,18,ØØ,C9,7E,23,FE,ØD,28
,Ø3,DF,18,F7,C9,E5,C5,CD,E4,ØØ,C1,E
1,38,ØF,77,23,1Ø,F3,C9,21,7E,8D
15Ø DATA 18,E3,21,8F,8D,18,DE,E1,21
,A1,8D,CD,2Ø,8D,C3,E7,ØØ,4E,6F,6D,6
2,72,65,3A,2Ø,2Ø,2Ø,2Ø,2Ø,2Ø,ØD
16Ø DATA 49,6E,69,63,69,6F,3A,ØD,46
,69,6E,61,6C,3A,ØD,45,6A,65,63,75,6
3,69,A2,6E,3A,ØD,4C,6F,6E,67,69
17Ø DATA 74,75,64,3A,ØD,5Ø,72,6F,67
,72,61,6D,61,2Ø,65,6E,2Ø,4C,4F,41,4
4,ØD,5Ø,72,6F,67,72,61,6D,61,2Ø
18Ø DATA 65,6E,2Ø,43,4C,4F,41,44,ØD
,45,72,72,6F,72,2C,63,61,72,67,61,2
Ø,69,6E,74,65,72,72,75,6D,7Ø,69
19Ø DATA 64,61,2Ø,6F,2Ø,64,65,66,69
,63,69,65,6E,74,65,ØD,ØØ,2Ø,2Ø,2Ø,2
Ø,2Ø,2Ø,2Ø,2Ø,11,1Ø,27,CD,EB,8D
2ØØ DATA 11,E8,Ø3,CD,EB,8D,11,64,ØØ
,CD,EB,8D,11,ØA,ØØ,CD,EB,8D,11,Ø1,Ø
Ø,3E,2F,19,3C,FE,ØØ,ED,52,4F,CD
21Ø DATA 2Ø,ØØ,79,3Ø,F4,DF,C9
```



```
; *** CASNOM ***
;
; Programa que averigua las
; direcciones de inicio, final
; y ejecución de un programa
; grabado en cinta .
;
; por Roni Van-Ginkel
;
```



PROGRAMAS

```
10                        ORG  36000
11                        LOAD 36000
12 8CA0 CDE100            CALL 0E1H
13 8CA3 CDE400            CALL 0E4H
14 8CA6 32C78D            LD   (DAT6),A
15 8CA9 CDE400    LOOP:   CALL 0E4H
16 8CAC FECF              CP   207
17 8CAE 30F9              JR   NC,LOOP
18 8CB0 21538D            LD   HL,DAT1
19 8CB3 77                LD   (HL),A
20 8CB4 23                INC  HL
21 8CB5 0605              LD   B,5
22 8CB7 CD2A8D            CALL READ
23 8CBA CDE700            CALL 0E7H
24 8CBD 214C8D            LD   HL,DAT0
25 8CC0 CD208D            CALL TEXT
26 8CC3 CD158D            CALL LF
27 8CC6 3AC78D            LD   A,(DAT6)
28 8CC9 FED3              CP   211
29 8CCB 2870              JR   Z,CLOAD
30 8CCD FEEA              CP   234
31 8CCF 2867              JR   Z,LOAD
32 8CD1 CDE100            CALL 0E1H
33 8CD4 21C88D            LD   HL,DAT7
34 8CD7 0606              LD   B,6
35 8CD9 CD2A8D            CALL READ
36 8CDC 01C88D            LD   BC,DAT7
37 8CDF 0A                LD   A,(BC)
38 8CE0 5F                LD   E,A
39 8CE1 03                INC  BC
40 8CE2 0A                LD   A,(BC)
41 8CE3 57                LD   D,A
42 8CE4 03                INC  BC
43 8CE5 0A                LD   A,(BC)
44 8CE6 6F                LD   L,A
45 8CE7 03                INC  BC
46 8CE8 0A                LD   A,(BC)
47 8CE9 67                LD   H,A
48 8CEA 03                INC  BC
49 8CEB 03                INC  BC
50 8CEC 03                INC  BC
51 8CED ED52              SBC  HL,DE
52 8CEF 7D                LD   A,L
53 8CF0 02                LD   (BC),A
54 8CF1 03                INC  BC
55 8CF2 7C                LD   A,H
56 8CF3 02                LD   (BC),A
57 8CF4 11C88D            LD   DE,DAT7
58 8CF7 215A8D            LD   HL,DAT2
59 8CFA 0604              LD   B,4
60 8CFC E5                PUSH HL
61 8CFD E1        DIREC:  POP  HL
62 8CFE CD208D            CALL TEXT
63 8D01 E5                PUSH HL
```



PROGRAMAS

```
8D02 1A                LD   A,(DE)
8D03 6F                LD   L,A
8D04 13                INC  DE
8D05 1A                LD   A,(DE)
8D06 67                LD   H,A
8D07 13                INC  DE
8D08 D5                PUSH DE
8D09 CDD08D            CALL PRINT
8D0C CD158D            CALL LF
8D0F D1                POP  DE
8D10 10EB              DJNZ DIREC
8D12 E1                POP  HL
8D13 1834              JR   FIN
8D15 3E0D      LF:     LD   A,13
8D17 CD1800            CALL 18H
8D1A 3E0A              LD   A,10
8D1C CD1800            CALL 18H
8D1F C9                RET
8D20 7E        TEXT:   LD   A,(HL)
8D21 23                INC  HL
8D22 FE0D              CP   13
8D24 2803              JR   Z,FTEXT
8D26 DF                RST  18H
8D27 18F7              JR   TEXT
8D29 C9        FTEXT:  RET
8D2A E5        READ:   PUSH HL
8D2B C5                PUSH BC
8D2C CDE400            CALL 0E4H
8D2F C1                POP  BC
8D30 E1                POP  HL
8D31 380F              JR   C,ERROR
8D33 77                LD   (HL),A
8D34 23                INC  HL
8D35 10F3              DJNZ READ
8D37 C9                RET
8D38 217E8D    LOAD:   LD   HL,DAT3
8D3B 18E3              JR   TEXT
8D3D 218F8D    CLOAD:  LD   HL,DAT4
8D40 18DE              JR   TEXT
8D42 E1        ERROR:  POP  HL
8D43 21A18D            LD   HL,DAT5
8D46 CD208D            CALL TEXT
8D49 C3E700    FIN:    JP   0E7H
8D4C 4E6F6D62  DAT0:   DEFB "Nombre:"
8D50 72653A
8D53 20202020  DAT1:   DEFB "      "
8D57 2020
8D59 0D                DEFB 13
8D5A 496E6963  DAT2:   DEFB "Inicio:"
8D5E 696F3A
8D61 0D                DEFB 13
8D62 46696E61          DEFB "Final:"
8D66 6C3A
8D68 0D                DEFB 13
8D69 456A6563          DEFB "Ejecución:"
8D6D 756369A2
8D71 6E3A
8D73 0D                DEFB 13
8D74 4C6F6E67          DEFB "Longitud:"
8D78 69747564
8D7C 3A
8D7D 0D                DEFB 13
8D7E 50726F67  DAT3:   DEFB "Programa en LOAD"
8D82 72616D61
8D86 20656E20
8D8A 4C4F4144
8D8E 0D                DEFB 13
8D8F 50726F67  DAT4:   DEFB "Programa en CLOAD
8D93 72616D61
8D97 20656E20
8D9B 434C4F41
8D9F 44
8DA0 0D                DEFB 13
8DA1 4572726F  DAT5:   DEFB "Error,carga "
8DA5 722C6361
8DA9 72676120
8DAD 696E7465          DEFB "interrumpida o "
8DB1 7272756D
8DB5 70696461
8DB9 206F20
8DBC 64656669          DEFB "deficiente"
8DC0 6369656E
8DC4 7465
8DC6 0D                DEFB 13
8DC7 00        DAT6:   DEFB 0
8DC8 20202020  DAT7:   DEFB "        "
8DCC 20202020
8DD0 111027    PRINT:  LD   DE,10000
8DD3 CDEB8D            CALL PR1
8DD6 11E803            LD   DE,1000
8DD9 CDEB8D            CALL PR1
8DDC 116400            LD   DE,100
8DDF CDEB8D            CALL PR1
8DE2 110A00            LD   DE,10
8DE5 CDEB8D            CALL PR1
8DE8 110100            LD   DE,1
8DEB 3E2F      PR1:    LD   A,47
8DED 19                ADD  HL,DE
8DEE 3C        PR2:    INC  A
8DEF FE00              CP   0
8DF1 ED52              SBC  HL,DE
8DF3 4F                LD   C,A
8DF4 CD2000            CALL 20H
8DF7 79                LD   A,C
8DF8 30F4              JR   NC,PR2
8DFA DF                RST  18H
8DFB C9                RET
                       END
```

PROGRAMAS

CARBUS

por Roni Van-Ginkel

Este corto programa sirve para buscar en una zona de memoria previamente especificada una cantidad determinada de datos (bytes).

Los datos que deseemos buscar pueden ser de 3 formatos:

ASCII: Escribimos los caracteres que queramos buscar y pulsamos RETURN.

BYTES: Decimos cuántos datos deseamos introducir e introducimos dichos datos en decimal.

DIRECCION: Introducimos la dirección que deseemos buscar.

El programa nos irá escribiendo todas las direcciones donde se encuentran los datos a buscar, cuando haya terminado (o no haya encontrado ninguna dirección) escribirá FIN y terminará.

El programa en Basic está lo más empaquetado posible para limitar la menor cantidad de memoria posible.

Aparte del listado en ensamblador, adjuntamos dos listados, el primero es el cargador de Basic, y el segundo es el listado en sí.

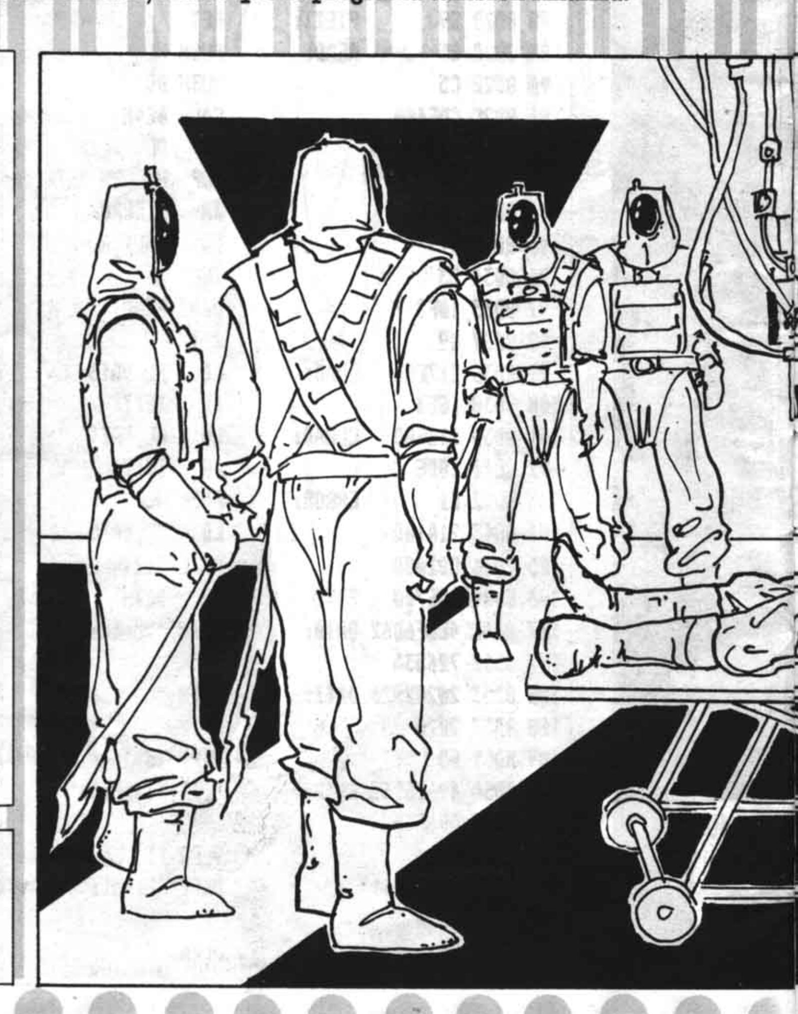
El listado en ensamblador es bastante corto, y se limita a coger de la tabla donde se encuentran los caracteres a buscar el primero, con la instrucción CPIR busca en la memoria el lugar donde se encuentra, y una vez encontrado, compara los demás datos. Como el programa guarda en la memoria la dirección donde se encuentra al volver al Basic, y la cantidad de memoria que le falta de explorar, al llamarlo de nuevo, seguirá explorando desde donde lo dejó.

El programa en Basic pone en la memoria todos los datos (dirección de inicio, longitud, caracteres a buscar y una señal de fina que es un 13 (un return en ASCII), por lo que queda limitado a buscar caracteres que no sean iguales a 13. Pero los que sepan un poco de programación no le encontrarán ningún problema a cambiar el código 13 por otro que les vaya mejor.

Una vez introducidos los parámetros, el programa va llamando a la subrutina de ensamblador hasta que ésta le devuelve un 0 como la dirección a buscar, indicando así que ya ha finalizado, con lo que el programa también finaliza.

```
1 '
2 ' *** Cargador CARBUS ***
3 '
10 CLS:PRINT"Un momento ..."
20 FOR I=34400! TO 34451!
30 READ A$:A=VAL(A$)
40 POKE I,A
50 N=N+A
60 NEXT
70 IF N<>1352 THEN PRINT"ERROR EN D
ATAS":STOP
80 PRINT"GRABAR (S/N)";
90 A$=INPUT$(1):PRINTA$:IF A$<>"S"
AND A$<>"N" THEN GOTO 80
100 IF A$="N" THEN END
110 BSAVE"CAS:CARBUS",34400!,34451!
120 END
130 DATA 2A,94,86,ED,4B,96,86,11,98
,86,1A,ED,B1,E5,78,FE,00,20,0B,79,F
E,00,20,06,E1,21,01,00,18,0E,13
140 DATA 1A,FE,0D,28,07,BE,23,28,F6
,E1,18,DC,E1,22,94,86,ED,43,96,86,C
9
```

```
10 CLEAR200,34400!:SCREEN0:KEYOFF:W
IDTH40:COLOR15,4,4:DEFUSR=34400!:IN
PUT"Direccion inicial:";A:INPUT"Dir
eccion final:";B:IFB<=ATHENB=65535!
```






```
:LOCATE17,1:PRINTB
20 PRINT "(A)scii (B)ytes (D
)ireccion :";:A$=INPUT$(1):P
RINT A$:ON ASC(A$)-64 GOTO 4
Ø,6Ø,2Ø,5Ø
3Ø GOTO2Ø
4Ø INPUT"Caracteres: ";A$:FO
RI=1TOLEN(A$):POKE34459!+I,A
SC(MID$(A$,I,1)):NEXT:GOTO7Ø
5Ø INPUT"Direccion:";C:D=INT
(C/256):E=C-D*256:POKE 3446Ø
!,E:POKE34461!,D:POKE 34462!
,13:GOTO8Ø
6Ø INPUT"Cuantos datos";C:FO
RI=1TOC:PRINT I;:INPUT "dato
: ";D:POKE 34459!+I,D:NEXT
7Ø POKE 34459!+I,13
8Ø PRINT:C=INT(A/256):D=A-C*
256:POKE34456!,D:POKE34457!,
C
9Ø G=B-A:E=INT(G/256):F=G-E*
256:POKE34458!,
!,F:POKE34459!,E:D=USR(Ø):A=PEEK(34456!)+
PEEK(34457!)*256-1:IFA=ØTHENPRINT"Fin":EN
DELSEPRINT"Direccion:";A:GOTO9Ø
1ØØ END
```

```
                        ; * * *  CARBUS  * * *
                        ;
                        ; Subrutina en ensamblador
                        ; para buscar una cadena de
                        ; bytes en la memoria y
                        ; devolver la dirección donde
                        ; se encuentra dicha cadena
                        ; al Basic .
                        ;
                        ; Por Roni Van-Ginkel
                        ;
                                ORG  34400      ; INICIO PROGRAMA
                                LOAD 34400
8660 2A9886                     LD   HL,(DAT1)  ;HL=DIR.INICIAL
8663 ED4B9A86                   LD   BC,(DAT2)  ;BC=LONGITUD
8667 119C86         CARG:       LD   DE,DAT3    ;DE=COMIENZO
866A 1A             LOOP:       LD   A,(DE)     ;A=PRIMER CARAC
866B EDB1                       CPIR            ;BUSCA
866D E5                         PUSH HL         ;HL=DIR. CARAC. +1
866E 78                         LD   A,B        ;VER SI BC=0
866F FE00                       CP   0
8671 200B                       JR   NZ,LOOP1
8673 79                         LD   A,C
8674 FE00                       CP   0
8676 2006                       JR   NZ,LOOP1
8678 E1                         POP  HL         ;SI LO ES, VOLVER
8679 210100                     LD   HL,1
867C 1812                       JR   FIN
867E 13             LOOP1:      INC  DE         ;BUCLE BUSQUEDA
867F 1A                         LD   A,(DE)
8680 FE0D                       CP   13         ;FINAL DE LA TABLA?
8682 280B                       JR   Z,FIN1
8684 BE                         CP   (HL)
8685 23                         INC  HL
8686 28F6                       JR   Z,LOOP1    ;SI COINCIDEN SIGUE
8688 FE23                       CP   35         ;ES UN COMODIN?
868A 28F2                       JR   Z,LOOP1
868C E1                         POP  HL         ;NO COINCIDEN, VUELVE
868D 18D8                       JR   CARG
868F E1             FIN1:       POP  HL         ;DIR. ENCONTRADA
8690 229886         FIN:        LD   (DAT1),HL  ;CARGA DIR.
8693 ED439A86                   LD   (DAT2),BC  ;CARGA LONGITUD
8697 C9                         RET
                                ORG  34456      ;ORIGEN DE LA TABLA
                                LOAD 34456
8698 0000           DAT1:       DEFB 0,0        ;DIR. INICIO
869A 0000           DAT2:       DEFB 0,0        ;LONGITUD
869C 00             DAT3:       DEFB 0          ;CADENA DE CARAC.
                                END
```



LETBUS

Por Roni Van-Ginkel

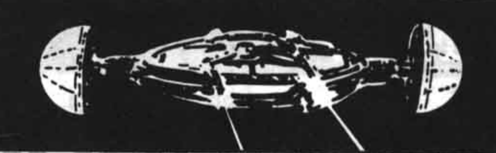
Este programa realizado en Ensamblador nos permitirá visualizar toda la pantalla en formato de texto, con lo cual podremos localizar cualquier mensaje de un programa, encontrar su comienzo o simplemente localizar las zonas de memoria donde se almacenan variables alfanuméricas.

El programa incluye a su vez una subrutina (la que comienza por el label PRINT) que nos permitirá imprimir el valor de HL en decimal en la pantalla.

```
1 '
2 ' Cargador para el programa LETBUS
3 '
10 CLS:PRINT"Un momento ..."
20 FOR I=36000! TO 36153!
30 READ A$
40 A=VAL("&H"+A$)
50 POKE I,A
60 N=N+A
70 NEXT
80 IF N<>15634! THEN PRINT"ERROR EN DATAS":STOP
90 PRINT "Para grabar hacer:":PRINT "BSAVE "+CHR$(34)+"LETBUS"+CHR$(34)+",36000,36155":END
100 DATA 3E,0F,32,E9,F3,3E,04,32,EA,F3,32,EB,F3,3E,28,32,AE,F3,CD,6C,00,CD,CC,00,21,00,80,11,28,00,01
110 DATA 98,03,E5,CD,5C,00,E1,3E,00,32,16,F4,E5,2E,01,26,11,CD,C6,00,E1,E5,CD,0E,8D,E1,3E,08,CD,41,01
120 DATA FE,FF,28,F7,CB,7F,28,12,CB,67,28,12,CB,77,28,19,CB,6F,28,0D,CB,47,28,17,18,C3,23,C3,BB,8C,2B
130 DATA 18,BC,11,27,00,ED,52,C3,BB,8C,11,28,00,19,18,AE,C9,11,10,27,CD,29,8D,11,E8,03,CD,29,8D,11,64
140 DATA 00,CD,29,8D,11,0A,00,CD,29,8D,11,01,00,3E,2F,19,3C,FE,00,ED,52,4F,CD,20,00,79,30,F4,DF,C9
```

```
        ; * * * LETBUS * * *
        ;
        ; Programa que presenta toda
        ; la memoria central en formato
        ; caracteres para localizar los
        ; textos de un programa .
        ;
        ; Por Roni Van-Ginkel
        ;
                ORG  32800              ;DIRECCION INICIO
```





PROGRAMAS

```
                         LOAD 32800
8020 3E0F                LD   A,15          ;COLOR 15,4,4
8022 32E9F3              LD   (0F3E9H),A
8025 3E04                LD   A,4
8027 32EAF3              LD   (0F3EAH),A
802A 32EBF3              LD   (0F3EBH),A
802D 3E28                LD   A,40
802F 32AEF3              LD   (0F3AEH),A
8032 CD6C00              CALL 6CH           ;SCREEN 0
8035 CDCC00              CALL 0CCH
8038 210080   PROG:      LD   HL,32768      ;HL=INICIO
803B 112800   BUCLE:     LD   DE,40         ;DE=ORG PANT
803E 019803              LD   BC,920        ;BC=LONGITUD
8041 E5                  PUSH HL
8042 CD5C00              CALL 5CH           ;PONER EN PANTALLA
8045 E1                  POP  HL
8046 3E00                LD   A,0           ;SALIDA POR PANTALLA
8048 3216F4              LD   (0F416H),A
804B E5                  PUSH HL
804C 2E01                LD   L,1
804E 2611                LD   H,17
8050 CDC600              CALL 0C6H
8053 E1                  POP  HL
8054 E5                  PUSH HL
8055 CD8E80              CALL PRINT
8058 E1                  POP  HL
8059 3E08     TECLA:     LD   A,8           ;TECLADO
805B CD4101              CALL 141H
805E FEFF                CP   255           ;PULSACION ?
8060 28F7                JR   Z,TECLA       ;SI NO ESPERA
8062 CB7F                BIT  7,A
8064 2812                JR   Z,RIGHT       ;CURSOR DERECHA
8066 CB67                BIT  4,A
8068 2812                JR   Z,LEFT        ;CURSOR IZQUIERDA
806A CB77                BIT  6,A
806C 2819                JR   Z,DOWN        ;CURSOR ABAJO
806E CB6F                BIT  5,A
8070 280D                JR   Z,UP          ;CURSOR ARRIBA
8072 CB47                BIT  0,A
8074 2817                JR   Z,EXIT        ;BARRA = SALIR
8076 18C3                JR   BUCLE
8078 23       RIGHT:     INC  HL            ;AVANZAR LENTO
8079 C33B80              JP   BUCLE
807C 2B       LEFT:      DEC  HL            ;RETROCEDER LENTO
807D 18BC                JR   BUCLE
807F 112700   UP:        LD   DE,39         ;RETROCEDER RAPIDO
8082 ED52                SBC  HL,DE
8084 C33B80              JP   BUCLE
8087 112800   DOWN:      LD   DE,40         ;AVANZAR RAPIDO
808A 19                  ADD  HL,DE
808B 18AE                JR   BUCLE
808D C9       EXIT:      RET                ;VOLVER AL BASIC
                         ;
                         ; SUBRUTINA QUE INPRIME EL
                         ; VALOR DE HL EN DECIMAL
                         ;
808E 111027   PRINT:     LD   DE,10000
8091 CDA980              CALL PR1
8094 11E803              LD   DE,1000
8097 CDA980              CALL PR1
809A 116400              LD   DE,100
809D CDA980              CALL PR1
80A0 110A00              LD   DE,10
80A3 CDA980              CALL PR1
80A6 110100              LD   DE,1
80A9 3E2F     PR1:       LD   A,47
80AB 19                  ADD  HL,DE
80AC 3C       PR2:       INC  A
80AD FE00                CP   0
80AF ED52                SBC  HL,DE
80B1 4F                  LD   C,A
80B2 CD2000              CALL 20H
80B5 79                  LD   A,C
80B6 30F4                JR   NC,PR2
80B8 DF                  RST  18H
80B9 C9                  RET
                         END
```

TRUCOS DEL PROGRAMADOR



SIMETRIA EN SCREEN 2

Javier Herreruela Morón, de Madrid, nos envía un corto pero interesante programa en ensamblador. Este programa permite la realización de una simetría de la pantalla respecto al eje vertical que pasa por el centro de la misma.

De este modo, la mitad izquierda de la pantalla resultará copiada simétricamente en la mitad derecha. Hay que avisar que esta simetría se realizará sólo si estamos en SCREEN 2.

La rutina en C.M. ocupa 54 bytes ubicados entre las direcciones 50000 y 50053. Sin embargo la rutina es totalmente reubicable y podéis situarla en la dirección que os parezca más oportuna.

Incluimos dos listados. El primero es el correspondiente al programa fuente en lenguaje ensamblador, mientras que el segundo es el cargador en BASIC. En este segundo listado hemos incluido una llamada a la rutina a modo de ejemplo, que podéis sustituir por vuestros propios gráficos.

La rutina tarda 2.5 segundos en ejecutarse totalmente. Aquí os incluimos los correspondientes listados.

LISTADO 2

```
                          ORG   50000

                VPEEK:    EQU   4AH
                VPOKE:    EQU   4DH

C350 210000               LD    HL,0
C353 CD4A00     PAT:      CALL  VPEEK
C356 0608                 LD    B,8
C358 CB3F       BYT:      SRL   A
C35A CB11                 RL    C
C35C 10FA                 DJNZ  BYT
C35E E5                   PUSH  HL
C35F 7D                   LD    A,L
C360 EEF8                 XOR   248
C362 6F                   LD    L,A
C363 79                   LD    A,C
C364 CD4D00               CALL  VPOKE
C367 E1                   POP   HL
C368 23                   INC   HL
C369 7C                   LD    A,H
C36A FE18                 CP    24
C36C 20E5                 JR    NZ,PAT
C36E 210020               LD    HL,8192
C371 CD4A00     COL:      CALL  VPEEK
C374 4F                   LD    C,A
C375 E5                   PUSH  HL
C376 7D                   LD    A,L
C377 EEF8                 XOR   248
C379 6F                   LD    L,A
C37A 79                   LD    A,C
C37B CD4D00               CALL  VPOKE
C37E E1                   POP   HL
C37F 23                   INC   HL
C380 7C                   LD    A,H
C381 FE38                 CP    56
C383 20EC                 JR    NZ,COL
C385 C9                   RET
                          END
```

LISTADO 1

```
1 '
2 ' *** SIMETRIA ***
3 '
4 ' Por Javier Herreruela
5 '
6 ' Para MSX-Extra
7 '
10 FOR J=50000! TO 50053!
20 READ A$:POKE J,VAL("&H"+A$)
30 S=S+PEEK(J):NEXT
40 IF S<>6295 THEN PRINT "Hay un ER
ROR":END
50 SCREEN 2
60 FOR H=1 TO 50:PSET (255*RND(-TIM
E),191*RND(1)),15*RND(1):NEXT
70 CIRCLE (50,90),30,,,,4/3
80 LINE (0,0)-(255,191)
90 BEEP:A$=INPUT$(1)
100 DEF USR=50000!:A=USR(0)
110 GOTO 110
1000 DATA 21,0,0,cd,4a,0,06,08,cb,3
f,cb,11,10,fa,e5,7d,ee,f8
1010 DATA 6f,79,cd,4d,0,e1,23,7c,fe
,18,20,e5,21,0,20
1020 DATA cd,4a,0,4f,e5,7d,ee,f8,6f
,79,cd,4d,0,e1,23,7c,fe,38,20,ec,c9
```



msxclub DE MAILING

¡NOS APLICAMOS A SER UTILES!

A TRAVES DE MSX CLUB DE MAILING PUEDES ADQUIRIR

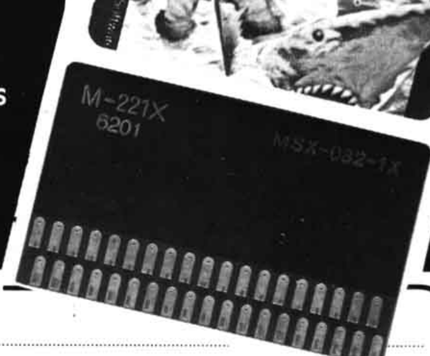
BASIC TUTOR IDEALOGIC



Deja el manual de lado. Inserta este breviario de BASIC en cartucho y olvídate. **No ocupa memoria.**
PVP 3.500 pts.

ADAPTADORES TARJETAS INTELIGENTES BEE CARD Y SOFTCARD





No te quedes al margen y disfruta de las tarjetas inteligentes. Lo último en soft.

ENVIA HOY MISMO ESTE CUPON

Nombre y apellidos ....

Dirección ....

Población .... CP .... Prov. .... Tel. ....

- ☐ Tutor Basic Ptas. 3.500,–
- ☐ Sweet Acorn Ptas. 5.200,–
- ☐ Barn Stormer Ptas. 5.200,–
- ☐ Adaptador Bee Card Ptas. 2.850,–
- ☐ Backgammon Ptas. 5.200,–
- ☐ Chock'n Pop Ptas. 5.200,–
- ☐ Adaptador Softcard Ptas. 2.850,–
- ☐ Shark Hunter Ptas. 5.200,–
- ☐ Le Mans 2 Ptas. 5.200,–

Gastos de envío por **cada producto 100,– pts.** Remito talón bancario de .... pts. **a la orden de Manhattan Transfer, S.A.** Enviar a MSX CLUB de MAILING, Roca i Batlle 10-12 bajos - 08023 Barcelona.


NOVEDADES KONAMI
DISTRIBUIDO EXCLUSIVAMENTE POR SERMA EN ESPAÑA.
VAMPIRE KILLER
Enfréntate a Drácula en esta espectacular aventura de Konami. Atraviesa el camino del Diablo, consigue las armas y poderes especiales y quizá puedas atravesar el castillo satánico y luchar con el Maestro de la Oscuridad.
En cada esquina habrá adversarios que te helarán la sangre; necesitarás lograr los medios de superarlos. Una decisión equivocada te hará fracasar...
Este cartucho ROM con un Mega lleno de acción, con los brillantes gráficos a que Konami nos tiene acostumbrados, es una aventura del principio al final.
VAMPIRE KILLER ™
PENGUIN ADVENTURE ™
GAME MASTER ™
ENJOY MORE!
GAMESMASTER
Gamesmaster es la única respuesta para esos juegos difíciles.
Posibilidades de ralentizar el movimiento e incluso congelar el juego, modificar la velocidad y etapas del juego.
Volcados de pantalla por impresora, pasa a cinta o a disco las máximas puntuaciones. Selecciona el número de jugadores y calcula las máximas puntuaciones.
En realidad, algo esencial para los fans de los juegos de Konami.
PENGUIN ADVENTURE
Guía a nuestro héroe Penguin por cuevas, mares y glaciares para devolver la normalidad a la Princesa Penguin y su reino.
Pelea con los Tiranosaurios y con montones de extraños enemigos utilizando los poderes comprados al Comerciante Esquimal.
Apuesta los peces en una máquina tragaperras para aumentar la puntuación y bucea por escenarios submarinos en un intento de restaurar el Paraiso Penguin.
Un juego lleno de acción con los gráficos que acostumbras esperar de Konami.
NUEVOS CARTUCHOS
Vampire Killer - 6.800 pts. (solo MSX 2)
Penguin Adventure - 6.150 pts.
Game Master - 6.150 pts.
KONAMI SOLO EDITA SUS PRODUCTOS PARA MSX EN CARTUCHO
SERMA
TITULO:
NOMBRE Y APELLIDOS:
DIRECCION:
COD. POSTAL:
POBLACION:
PROVINCIA:
FORMA DE PAGO: CONTRARREMBOLSO ☐ POR TALON BANCARIO ☐
RECORTA Y ENVIA ESTE CUPON A KONAMI SHOP.
FRANCISCO NAVACERRADA, 19. 28028 MADRID.